DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5002

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

A5657

DESIGN AND IMPLEMENTATION OF THE PMS MODULE FOR "ARGOS"

by

Dionisios G. Antonopoulos

and

Henry V. Turner

December 1989

Thesis Advisor: C. Thomas Wu

Approved for public release; distribution is unlimited

T247163

Unclassified
Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification UNCLASSIFIED
1b Restrictive Markings
2a Security Classification Authority
3 Distribution Availability of Report
Approved for public release; distribution is unlimited.
2b Declassification/Downgrading Schedule
4 Performing Organization Report Number(s)
5 Monitoring Organization Report Number(s)
6a Name of Performing Organization
Naval Postgraduate School
6b Office Symbol (*If Applicable*) 37
7a Name of Monitoring Organization
Naval Postgraduate School
6c Address (*city, state, and ZIP code*)
Monterey, CA 93943-5000
7b Address (*city, state, and ZIP code*)
Monterey, CA 93943-5000
8a Name of Funding/Sponsoring Organization
8b Office Symbol (*If Applicable*)
9 Procurement Instrument Identification Number
8c Address (*city, state, and ZIP code*)
10 Source of Funding Numbers

Program Element Number	Project No	Task No	Work Unit Accession No

11 Title (*Include Security Classification*) DESIGN AND IMPLEMENTATION OF THE PMS MODULE FOR "ARGOS"
12 Personal Author(s) Antonopoulos, Dionisios G., Turner, Henry V.
13a Type of Report
Master's Thesis
13b Time Covered
From To
14 Date of Report (*year, month,day*)
1989 December
15 Page Count
273
16 Supplementary Notation The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.
17 Cosati Codes

Field	Group	Subgroup

18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Software Engineering, Event Driven Multimedia Database

19 Abstract (*continue on reverse if necessary and identify by block number*

Argos is a prototype multimedia database developed as a Battle Group Commander's assesment tool and a shipboard data management tool. The ultimate goal of Argos is to provide database support for the "Paperless Ship" Navy. This thesis furthers research on Argos by analyzing, designing, and implementing a fully workable Planned Maintenance System(PMS). This implementation demonstrates both the capabilities and benefits such a system would have for the navy.

20 Distribution/Availability of Abstract
☒ unclassified/unlimited ☐ same as report ☐ DTIC users
21 Abstract Security Classification
Unclassified
22a Name of Responsible Individual
C.T. Wu
22b Telephone (*Include Area code*)
(408) 646-3391
22c Office Symbol
52Wq
DD FORM 1473, 84 MAR
83 APR edition may be used until exhausted
All other editions are obsolete
security classification of this page
Unclassified

Approved for public release; distribution is unlimited.

Design and Implementation of the PMS Module for "Argos"

by

Dionisios Antonopoulos
Lieutenant , Hellenic Navy
B.A., Greek Naval Academy, 1979

and

Henry V. Turner
Lieutenant , United States Navy
B.S., University of Washington, 1982

Submitted in partial fulfillment of the requirements for
the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1989

ABSTRACT

Argos is a prototype multimedia database, developed as a Battle Group Commander's assesment tool and a shipboard data management tool. The ultimate goal of Argos is to provide database support for the "Paperless Ship" Navy. This thesis furthers research on Argos by analyzing, designing, and implementing a fully workable Planned Maintenance System (PMS). This implementation demonstrates both the capabilities and benefits such a system would have for the Navy.

c.1

TABLE OF CONTENTS

I. INTRODUCTION 1

II. THE PROBLEM STATEMENT 5

III. THE PROGRAMMING ENVIRONMENT 10

IV. IMPLEMENTATION OF THE PMS MODULE 15

- A. BACKGROUND 15
- B. MODULE DESIGN 15
- C. MODULE ACCESS 17
- D. THE PMS MODULE 17
  - 1. PMS_driver 17
    - a. General description 17
    - b. The PMS menu 19
    - c. The division menus 25
  - 2. The PMS Toolbox 26
    - a. General description 26
    - b. The PMS Toolbox Tools 26
  - 3. The "Planning" stack 30
    - a. General description 30
    - b. Scheduling Process 31
  - 4. The "Movies" stack 34
    - a. General description 34
  - 5. The "Password" stack 35
    - a. General description 35
  - 6. The "Daily Activity" stack 38
    - a. The "Automatic Job Scheduling Status Card" 38
  - 7. The "Daily Activities Card" 41
  - 8. The "Ship Data" stack 42
  - 9. The "Log" stack 44
  - 10. The "PMS" stack 45

[illegible] LIBRARY
[illegible] SCHOOL
[illegible] 93943-5

E. SUMMARY 46
V. CONCLUSIONS 47
APPENDIX A PMS USER'S MANUAL 50
APPENDIX B PMS MODULE SCRIPTS 147
LIST OF REFERENCES 261
BIBLIOGRAPHY 262
INITIAL DISTRIBUTION LIST 263

LIST OF FIGURES

Figure 1 "PMS Title Card" 18
Figure 2 "PMS Driver Card" 19
Figure 3 "Password Selection" 20
Figure 4 "Schedule Selection" 21
Figure 5 "Reports and Messages" 22
Figure 6 "Bar Coded Label" 22
Figure 7 "Spot Checks Selection" 23
Figure 8 "Go Selection" 23
Figure 9 "Print Selection" 24
Figure 10 "Tool Box Selection" 25
Figure 11 "Equipment Log Selection" 25
Figure 12 "Division Menus" 26
Figure 13 "Add a MIP Card" 27
Figure 14 "File to Stack Card" 27
Figure 15 "Empty Pms Db Card" 28
Figure 16 "Delete a MIP/MRC Card" 28
Figure 17 "MIP/MRC List Card" 29
Figure 18 "MIP,WC Responsibility Card" 30
Figure 19 "Ship's Activities Card" 31
Figure 20 "Movie Card" 34
Figure 21 "Personal Access Info Card, General" 35
Figure 22 "Personal Access Info Card,WC" 36
Figure 23 "Personal Access Info Card, Equip" 36
Figure 24 "Personal Access Info Card, Da" 37
Figure 25 "Personal Access Info Card, Toolbox" 37
Figure 26 "Scheduling Status Card" 39
Figure 27 "DAY/MH Distribution Field, List View" 40
Figure 28 "DAY/MH Distribution Field, Graph View" 41
Figure 29 "Daily Activities Card" 42

Figure 30 "Ship's Data Card" 43
Figure 31 "MIP Responsibility Card" 43
Figure 32 "Personnel Work Center Card" 44
Figure 33 "Machinery History Card" 44
Figure 34 "MIP Card" 45
Figure 35 "MRC Card" 45

ACKNOWLEDGMENTS

We would like to acknowledge the following individuals for the use of their external command and functions in our thesis.

- Will Cate for his ShutDown Xcmd
- Brad J. Hicks for his Copy File Xfcn
- Steve Mailer, Apple Computer, for his FileName Xfcn
- Guy de Piccitotto for his Hpopupmenu Xfcn

We would like to thank the following individual for the use of his shareware program in the preparation of our thesis.

- Raymond Lau , StuffIt version 1.5.1

Finally, we would like to thank Apple Computer Inc. We would not have been able to complete this thesis without the use of the Macintosh and Hypercard.

I. INTRODUCTION

VADM J. Metcalf, USN (Retired), is credited with the concept of the "paperless ship" navy. He expressed the idea of eliminating all material held in paper form from the navy's ships. Today, vast amounts of paper are required in the day to day operation of the navy. This paper consists of many different types of documents and forms, including correspondence, personal records, medical records, pay records, technical manuals, and maintenance manuals. Substantial amounts of time and money are required to maintain this material, and the amount of weight and space that is taken up on ship is considerable. These resources can be put to better use for either weapons systems or to improve living conditions on board ship.

The current thinking on solving this problem revolves mostly around the transformation of information currently held in paper form to a digitized format for use with computers. However, the solutions being worked upon generally involve systems or databases that have been on the market for a while. Although many of these systems and databases are very capable, they don't necessarily provide the best solution because the technology involved is usually already outdated. For example, most databases currently in use don't allow for the integration of text and graphics. Additionally, the majority of equipment that the navy currently has is not generally suitable for presenting information in a natural form that personnel are used to seeing it in, i. e., text and graphics.

In January of 1988 a new solution was presented by two students at the Naval Post Graduate School at Monterey California. CDR B. B. Giannotti and Lt Kevin F. Duffy came up with the idea of developing a system using off the shelf equipment not usually associated with the navy [Ref. 1]. Their system, a multimedia database, simultaneously combines the use of several media at once. The media is in the form of graphic images, text, and sound (digitally recorded and computer generated). The name of their project is Argos. The purpose of their research was to provide a prototype system that could be used both as a Battle Group Commander's assessment tool and a shipboard data management tool.

The Apple Macintosh[1] was chosen as the equipment for the implementation of their prototype. This Macintosh was chosen because it has the ability to present information in multiple forms, including text, graphics, and sound. This capability exists without any requirement for any special programs or modifications to its internal components. Hypercard was chosen as the development environment for the system. Hypercard's programming language, Hypertalk, uses objects, techniques, and has many of the properties found in object-oriented languages such as Lisp and SmallTalk. Hypercard developed with all users of the Macintosh in mind, supports ease of development, modular programming and reusability. One added benefit of Hypercard is that it is delivered with every Macintosh, so there is no added cost for procurement.

Giannotti and Duffy divided their prototype into six different functional areas. These included: maintenance, operations, medical, supply, and administration. Their prototype consists of the development of only the maintenance area, specifically the LM2500 gas turbine engine on a FFG-7 class ship. This opened up many of the other areas for future research, development, and implementation.

---

[1] Macintosh, Hypercard, and HyperTalk are registered trademarks of Apple Computer.

The purpose of this thesis is to continue development on the Argos project. The Planned Maintenance System (PMS) section of the maintenance module was the area chosen for design and implementation .

PMS is the system developed by the navy to provide each ship, department, and supervisor with the tools needed to plan, schedule, and control planned maintenance. The system, as currently implemented, provides comprehensive procedures for planned maintenance, scheduling and control of tasks, and descriptions of the methods, materials, tools, and personnel needed for maintenance. Although the Planned Maintenance System (PMS) now in use has been very effective, it is believed that there is a better way of doing it. The system has changed little from when it was first introduced into the fleet. Although computers have become an everyday part of the navy, their involvement in the day to day operation of the Planned Maintenance System is almost nonexistent. There is one notable exception to this. The navy maintains an on-line database that is a repository for the complete list of MIPs and associated MRCs for each of the class of ships in commission. However, this database is not readily available to the average PMS user and is difficult to work with. Additionally, the information presented here consists only of the written sections of the cards with no provision for the presentation of any graphics or pictures.

The Planned Maintenance System as it now stands is tedious, time consuming, and man power intensive. The prototype system, "Argos", can be extended to bring the Planned Maintenance System up to date. Using the Macintosh and Hypercard, system can be designed and implemented that will bring relief from such mundane tasks as PMS scheduling, record keeping, spot checks and accomplishment tracking.

The purpose of this thesis is to provide a complete working PMS system, usable on any ship in the fleet. Although this system will work for any ship, the information contained in the database is relevant only to the FFG-7 class of ships. This is due to the time and resource constraints placed on the development of this system. All information in the database is up to date, as the information for the cards was down-loaded directly from the navy's PMS database.

The organization of the thesis is as follows. Chapter II, the problem statement, discusses the problems found in the PMS system as it currently implemented. Chapter III is a brief discussion of Hypercard and its programming language Hypertalk. Chapter IV discusses the approach taken in, and implementation details of , the system as developed. Chapter V provides conclusions, other programming environments looked at, and recommendations for follow-on work. The appendices consists of a users manual (detailed description of system use), and Hypertalk scripts for the various stacks of the system.

II. THE PROBLEM STATEMENT

In this day and age, computers have become a indispensable part of everyday life. They are found at almost every level of the Navy and are involved in just about every facet of Navy life. They have become integral parts of radar systems, fire control systems, and navigation systems. They are involved in payroll systems, satellite communications, ship's propulsion systems, and aircraft flight control systems. Personal computers, once a novelty item, can now be found in just about every ship's office.

Computers have been used to relieve the drudgery of every day life. They have been used to automate many of the mundane day to day tasks found throughout a ship or office. Individuals that have had the opportunity to work with computers have found out indispensable they are.

The Navy's goal for the future is to completely computerize every conceivable operation there is today. In that endeavor, the Navy has or is spending vast sums of money on research into computer systems and computer software.

Although the U. S. Navy is putting a great amount of effort and time into this goal, many operations on board ship are still being done with little or no help from computers. One such operation is the planning and scheduling of day to day maintenance on equipment found on board ship as provided through the Planned Maintenance System.

PMS is the system that was developed to provide each ship or department with the necessary tools for ensuring the maintainability of systems and equipment found on board ship. It is a system that has improved both the reliability and maintainability of the equipment. It has lead to improved levels of training of shipboard personnel because individuals get involved with the equipment on a daily basis. Long term costs involved with the repair of equipment have also been reduced due to the Planned Maintenance System. However, PMS is not without its problems.

PMS management as it is now implemented is very manpower intensive. Considerable amounts of paper work are generated and large amounts of time are spent on jobs which are routine and could be done on a computer. Other problems of PMS include:

- In daily operations, feed back information and record keeping can barely be taken in account because of the volumes of the information that is generated.
- An integrated view of the overall PMS system for a particular division is difficult to picture.
- History logs of individual equipments are usually unavailable.
- Planning of PMS jobs is a difficult task because in order to make the best utilization of available personnel one must take into account the ship's obligations and other special scheduling requirements which may or may not be available at the department or division level.
- Comments and other information related to the PMS jobs should stay together with the job in order to help as evaluate them and take actions to improve the overall maintenance system.
- Rescheduling of jobs, that were not completed on their assigned days, is also very difficult. Other maintenance jobs completely separate from PMS must also be included in the rescheduling in order to avoid overloading of any one particular work day.
- Time is waisted each day preparing for the days work assignments. Automatic generation of the daily work schedule for an entire quarter for each work center would give us the ability to prepare early for the next days and start jobs right away each morning without wasting time on preparation.
- PMS cards can be easily lost or destroyed, especially in engineering compartments. Interested individuals are discouraged or prevented from taking cards and reading it at their own leisure. Instructions on the cards are only presented with text and diagrams that barely help the individual.

The goal of this thesis is to provide a solution to the problems encountered by maintenance personnel in the Navy. Through the use of computers, the overall Planned Maintenance System will be easier to work with. Computers will allow accessibility to all hands, which is going to lead to an improvement in both the condition of the equipment and the level of training of the personnel on board ships.

The problems for which solutions will be provided for in the design and implementation of the PMS module include the following.

Scheduling of the PMS jobs is a difficult task that up to now has been done completely by hand. Taking in consideration the various ship's obligations and activities with an equal weighing of every day's schedule is hard to do. Scheduling, as it is now done, involves little or no coordination between divisions and as a consequent of this, the fact that jobs require coordination between work centers is not taken into account. The individual preparing the schedule should have some knowledge of the ship's schedule for the next quarter, something which is not always available. The person assigning the repair maintenance jobs is usually not able to take in account the actual work load of a particular day because he doesn't have a cumulative man/hour visualization. The complete scheduling process is time consuming and very tedious, requiring the filling out of long term, quarterly, weekly and daily forms. The solution to this problem will encompass bypassing all this work and producing directly the daily schedule for each work center.

Rescheduling of the PMS jobs, though a task that is very important, is also very cumbersome. Supervisor's usually end up by rescheduling jobs that are not very important or leaving out jobs that should be rescheduled and completed as soon as possible. Not all scheduled jobs can always be done, supervisor's can end up completing jobs delayed only for a couple of days and leaving jobs delayed for longer amounts of time to be rescheduled again. The solution to this problem will encompass the automatic rescheduling of uncompleted jobs.

Classification of PMS jobs was not available up to now. Whether a job could be done under way or at port or under other special conditions was left up to the individual's common sense and knowledge. The possibility exists for supervisor's to end up planning propeller removals during periods the ship is under way, or major repair jobs on holidays during which the required personnel wouldn't be available to complete them. Job classification should automatically be taken into account during job scheduling.

Naval ships can only carry a limited amount of copies of the PMS cards. During the performance of very long jobs, in hard to reach spaces, it is desired that the individual who completes them to have a copy of the PMS card along with him in order to perform it step by step with precision. Cards can be lost or destroyed during use. The ship as a self contained operating unit should have the capability to reproduce its own PMS cards. All the on-line PMS systems developed up to now consist only of static text for reference purposes only. The solution to this problem is to provide a system that contains the actual PMS cards. Instructions to the personnel on completing a particular job should be given on an individual basis, and although the PMS cards are very well written, the instructions are not always self explanatory to inexperienced personnel. The solution is provide for cards that are in a format that personnel are used to seeing. The cards will include diagrams, tables, extensive search capabilities, pictures showing the hard to identify parts of individual equipments, and instructive photographs of personnel doing the PMS jobs.

Historical status logs for equipments are something very hard to keep track of. Important PMS jobs should stay together with other repair jobs in a 'history card' for each piece of equipment, so that a record of equipment performance and required maintenance can be used to prevent future failures.

Data on a particular piece of equipment is hard to find, because of the various bulky volumes must be referred to in order to gather all the pertinent information.

Reports and other comparative feed back information are not easy to see in the current formats.

All notes and comments about any one PMS job should stay together with that job so that the supervisor can easily review them for evaluation purposes.

Repair parts needed for a PMS job should be easily accessible directly from the job and cumulative catalogs for all the repair parts should be generated. This will save both time and work, by having the repair parts ready and on the spot before beginning working on a PMS job.

The previously mentioned problems are by no means the only ones, but they represent the ones that are the most tedious and time consuming to do on a day to day basis.

III. THE PROGRAMMING ENVIRONMENT

A. BACKGROUND

Hypercard is a completely self contained environment for the development of information software. It is a software construction set consisting of stacks, fields, buttons, cards, images, and backgrounds that can be controlled, manipulated, and programmed. It gives the user the ability to create information based software for the Macintosh without extensive programming.

Up until now, the data base management system has been the cornerstone of information management on computers. File management and relational database management are but two of these.

File management is the simplest form of database software. It allows the collection of related information into a simple file. Simple search capabilities exist, but related information contained in separate files cannot be linked together. In other words, duplication of information is inherent in this system

Relational databases are the more sophisticated form of DBMS. The relational system allows links to be developed between related information contained in different files. The capabilities of sorting, searching, and retrieving of information also exists, but information can simultaneously be retrieved from multiple files where the desired information is related. Because of this, information is not duplicated.

B. HYPERCARD

The background information on the DBMS was an important prerequisite to talking about Hypercard because the two may appear to do the same kinds of things.

Hypercard is much like the relational database management system in that it has the capability to link related information contained in different stacks. HyperCard, however, allows the delivering of information in terms that go beyond the methods found in conventional databases. There are no constraints when searching for information, because there is no predetermined path to follow to retrieve related facts about a subject. Unlike a relational database which is only capable of retrieving information, Hypercard allows the user to move between stacks as the information needs require. It goes one step farther, instead of only retrieving the desired information, Hypercard allows the user to view the full context of related information.

Hypercard allows documents and pictures to be linked through buttons, fields, and user defined commands to other documents and pictures. These links, enabled by a mouse or other input device, allow the user to instantly branch to related facts as information is needed and retrieve them .

Hypercard has three stages of user interface/interaction [Ref 2]. The first is the browsing/typing level which restricts the users to finding and entering information in existing applications. The second is the painting/authoring level which allows the user to design and produce simple applications. At this level, the design tools are available to the user to modify existing applications or create new ones. The final level is the programming level. At this level, the user can modify the performance of existing applications or create new ones from scratch. This level is built around Hypertalk.

1. HyperTalk

Hypertalk is Hypercard's scripting (programming) language. Its roots can be found in object oriented languages. However, it is set apart from other languages in many ways.

Hypertalk's vocabulary is much like real English. Its syntax rules are some of the most flexible found in any computer language. It is not only forgiving of things like extra spaces but also anticipates multiple ways for expressing a command. Its grammar, unlike other languages, requires no elaborate statements but instead allows the user to write scripts that are easy to understand the first time they are read. Scripts can be written to perform a variety of tasks, including information look-up, any Hypercard menu command, running another application, or retrieving information about conditions within the computer.

Hypertalk uses objects and many of the techniques like those found in object-oriented programming. Object-oriented languages start with a core of objects from which the programmer can create descendant objects. The new objects inherit some or all of the properties of the ancestors but can add some of their own. HyperTalk is also based on a set of objects consisting of buttons, fields, cards, and backgrounds that are created in Hypercard, unlike the inheritance described above, Hypercard's descendant objects are are already predefined from the Hypercard object and can't be changed. Since objects are already defined, this allows for a much simpler view of how the object-oriented system works.

Hypertalk is event driven. Events are caused by the keyboard, mouse or a variety devices. Mouseup, mousedown, opencard, idle, and quit are just a few of these events. Scripts within objects are designed to react to these events. Hypercard is based on a hierarchy of objects. Hypercard has seven different levels, each of which corresponds to an object. When a event is received or a message signalling the event is received from an object like a button, if the button contains a script that will trap the event, then the script will be executed. If not then then the event will passed up the hierarchy of objects until it reaches a level where it is handled or is finally absorbed by the system.

HyperTalk allows a high degree modular programming because the functionality of a stack is spread throughout the many objects that make up the stack. Each object contains just a short script to produce the desired result when effected by a user or system generated message. Modularity makes it easier to test each script when designing the interaction between various objects and it ensures that features can be added at a later time without having to rewrite or modify the entire program.

Hypertalk has many of the same features found in procedural languages such as C/Pascal, including control structures, parameter passing functions, user-defined subroutines, string manipulation commands, data structures, graphics, sound commands and disk file routines.

HyperTalk, as it stands, provides a pretty complete vocabulary. However, when Hypercard was envisioned, it was decided that if needed, the vocabulary could be extended. In other words, a programmer can extend the functionality of Hypercard through the use of outside resources. This extendibility was provided by allowing external resources in the form of external commands and functions to be added to the system as the need arose. External commands are used to provide a special purpose command or function that cannot be done with the current set of HyperTalk commands. External commands are written in a language such as Pascal, C, or assembler. The value of external commands is that they allow for unlimited extendibility of Hypercard.

C. SUMMARY

Hypercard is not meant to replace the traditional DBMS's but as a way of enhancing the presentation of information. Hypercard can take a large database consisting of lines of text and dress it up by creating a graphical interface with controls to make accessing the information more intuitive and friendly.

Although Hypercard does have a few limitations such as having fixed size cards (this limitation is being addressed in the next version of Hypercard), its value as a programming environment can primarily be seen in that it offers an object-environment unlike any other in existence.

IV. IMPLEMENTATION OF THE PMS MODULE

A. BACKGROUND

In designing the PMS module, many decisions were made. The first was to provide a system that would be fully usable at the completion of the project. The second was to provide a user interface that matched as closely as possible that designed by the original architects of the Argos system. The third was that the module be usable in two different modes. The system should be usable as an integral part of Argos, accessible with little or knowledge to the user that the PMS module is a separate module from the main Argos system. The system should also be usable as a stand alone application. This decision was based on the judgement that in the day to day use of the PMS module, a typical user in maintenance would not need or actually have access to the full capabilities of the Argos system . The fourth was that the system should be password protected at all levels. It was decided that not all users of the system need have access to the full functionality of the system. A user's level of access should be based on the requirements of their position.

B. MODULE DESIGN

The design of the module is based on observations seen during research into the current Planned Maintenance System. Our observations provided us with the necessary information to define in the system the functions that we deemed are necessary for a functionally operating system. First, we noted that one of the most important aspects of the current system is the database of MIP/MRC cards. The current system has a heavy reliance on the cards but does not provide an efficient way of accessing, updating, or replacing them. Our design provides the necessary functionality to address these needs. Second, we observed that much of the work currently present in the system is being duplicated from division to division. In particularly, we observed that the process of job scheduling is done at the division level when it should be done at the ship level. We felt that for scheduling to be efficient , concurrent scheduling must take place between all divisions. Our design provides the necessary functionality to address this need, ensuring that work is not duplicated. Specifically, many of the functions currently done by multiple individuals will now be taken care of by one individual, the ship's PMS coordinator. Third, we noted that many of the methods involved in maintaining records on job completion and equipment accountability need updating. Our design provides the necessary functionality to address these needs.

The PMS module was divided into two groups of stacks. Dividing the system into individuals stacks ensured that as the system matured, greater functionality of the system could be appended into the system without rewriting the entire module.

The decision was made to divide the stacks into two different categories. The first are stacks that fit into the day to day operation of the system and as such provide the major functions of the system. The second are stacks that are needed to produce reports and generate new cards, but as such do not provide any other functionality to the system

The operational group of stacks consists of the following. The PMS_driver, Da stacks, PMS toolbox, Planning stack, PMS db, Movies db, Log db, Ship data db, Password db, and General label storage. The auxiliary group of stacks consists of the following. The Log_report, DA_report, MRC_print, Mip_printed, Comp_report, Generic_fld_print, Spot checks, and Card templates.

C. MODULE ACCESS

When the PMS module is used a part of the Argos system, two methods of accessing the module is provided. The first method is used when the user is at an upper level of the Argos system , still navigating through the ship, and has yet to descend down to one of the machinery levels. This method takes the user directly to the main driver of the module. Once at the driver level, the user has access to the full capabilities of the system, based on their access level and is presented with a list of options of which the system is capable of doing. The second method is used when the user is down at one of the machinery levels in Argos. When the user is at this level, selecting the PMS module will take the user directly to the respective MRC card for the piece of equipment that the user is currently at. If more complete access to this system is needed then the user must navigate up to one of the non-machinery levels.

When used as a stand alone application, at start-up, the user is initially at the driver level (Figure 1) with access to the functions of the system.

D. THE PMS MODULE

The following is a general description of the PMS module as implemented. If a more detailed description of use of the system is needed, refer to the PMS user's manual in Appendix A.

1. PMS_driver

a. General description

The PMS driver (Figure 2) is the main entry point into the PMS system. It is used to control and restrict access, depending on a user's access level, to the various other stacks. The PMS_driver provides the complete list of functions available to a user. Included in this operations such as viewing PMS cards, doing system spot checks and system scheduling.

Figure 1 PMS Title Card

One person per ship, the "SuperUser", has accessibility to the complete set of functions of the PMS module. In general, this person will normally be the PMS coordinator of the ship. The SuperUser has the responsibility of setting up each individual's access to the PMS system. Typically, an entire ship's crew should have access to the system, but access will vary according to their responsibilities and job assignment. For example, an individual who belongs to the EM01 work center should only have access to this work center. However, the engineering officer should have access to all engineering work centers. He, in turn would not have access to another division's work centers. Limiting an individual's access to the system ensures that the system is safeguarded from unwanted intrusion.

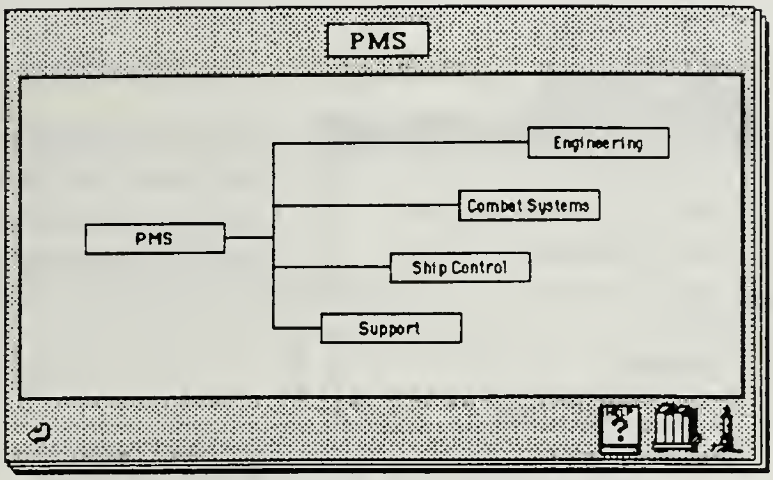


Figure 2 PMS Driver Card

Using the PMS menu, the user can perform the various functions of the PMS system. The performance of functions at this level does not preclude these same functions being available and used separately in the various other stacks. For example, the print function for a MRC card is available both at the card level and at the driver level. This dual functionality eliminates the process of searching for individual cards and then printing it at the card level.

The division menus provide the user with access, if access is given, to the current daily activity card for the work center selected. Throughout the system, functions not available to the user appear dimmed and therefore cannot be selected.

b. The PMS menu

The PMS menu is a two level menu system. A selection in level 1 provides additional choices that are relevant only to that particular selection.

The password choice (Figure 3) is used when setting up or changing an individual's access to the PMS module. Upon selection of this choice, the user is provided with additional selections to either add or remove a user from the system, to change their access, or to change an individual's access level.

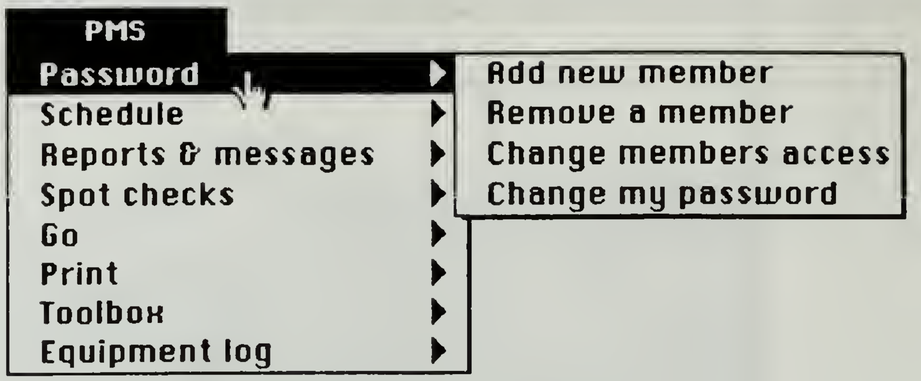


Figure 3 Password Selection

The selection, Add a new member, is available only to the SuperUser of the system. Upon selection, a new entry is automatically generated in the password database. At this point the SuperUser is presented with access card for the individual to be added to the system. The SuperUser uses this card to record pertinent information on the individual and then to set access to the various functions of the system. The generic password "me" is automatically given to the individual for their initial will remain so until changed by the user.

The selection, remove a member, is also only available to the SuperUser. It is used to remove a person from the system.

The selection, change member's access, is also available only to the SuperUser. It is used to change a user's access to the various functions of the system. This would be typically used in cases where there is a change in a user's responsibilities or assignment on board ship. Changing a person's access is done through the same procedure that is used for the add a member selection. After the name of the individual is entered into the system, their personal access card is shown on the screen, whereby the SuperUser can then make the necessary changes.

The selection, change my password, is available to every user that already has access to the PMS module. The user can select this choice at any time to change their password.

The selection, schedule (Figure 4), is used in the scheduling of the jobs for the PMS module.

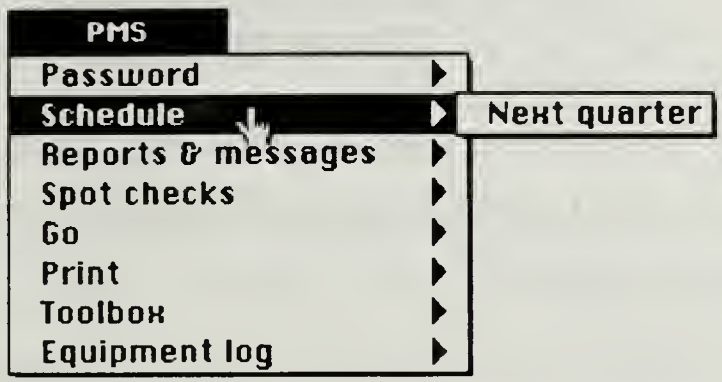


Figure 4 Schedule Selection

The selection, next quarter, is available only to the SuperUser of the system. It is used to initiate the automatic scheduling of PMS jobs for the entire ship. When selected, a set of calendar for the next quarter is generated and is then shown, one by one, on the screen to the SuperUser. Using this calendar, the SuperUser can set up the ship's schedule for the quarter. The system takes into account the ship's status for each day of the quarter, when the scheduling is started. It uses this information when deciding whether a particular day is suitable for the assigning of PMS jobs .

The selection, reports and messages (Figure 5), is used to print reports and/or other computer generated forms. Only the most basic of forms is provided, but as the need arises, other forms can be included .

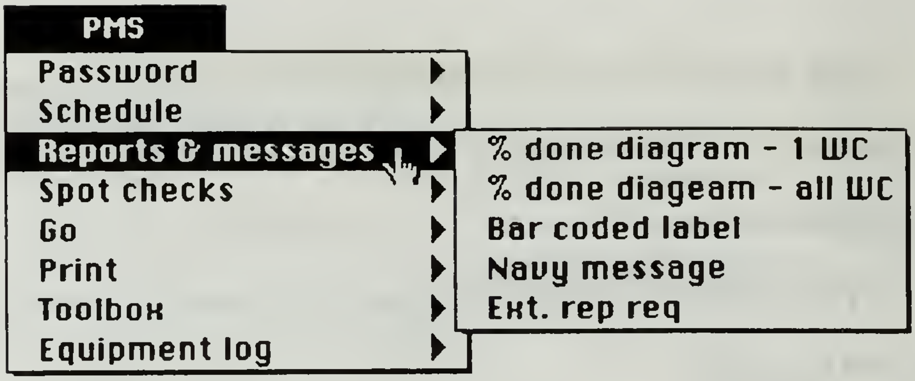
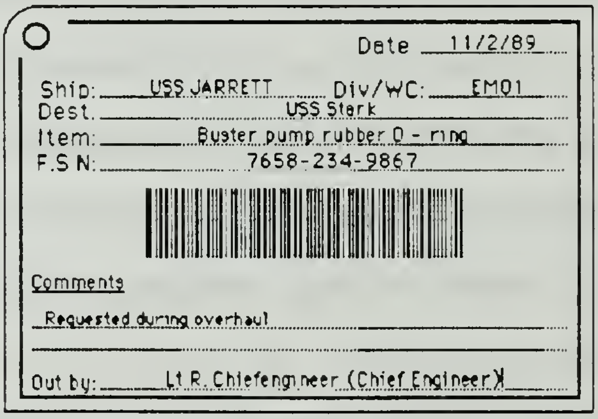


Figure 5 Reports and Messages

The percent done diagram selection is used to print out a report showing the percentage of PMS jobs that were completed for either one or all work centers on a particular day.

The selection, bar coded label, is used to produce a bar coded label to place on a piece of equipment that has to be removed from the ship for some reason. Specifically, if a piece of equipment has to go out for repair, be returned as destroyed, or is loaned to another ship, an accompanying label (Figure 6) is produced. It is used in identifying and tracking of all equipment on board ship.



Date 11/2/89

Ship: USS JARRETT Div/WC: EM01

Dest. USS Stark

Item: Buster pump rubber O - ring

F.S N: 7658-234-9867

Comments

Requested during overhaul

Out by: Lt R. Chiefengineer (Chief Engineer)

Figure 6 Bar Coded Label

The selection, spot check (Figure 7), allows the system to select, at random, PMS jobs that have been completed from the quarters' daily activities cards. A spot check sheet is generated automatically for a selected work center if "For one WC" is selected or for all the ships' work centers if "For entire ship" is selected.

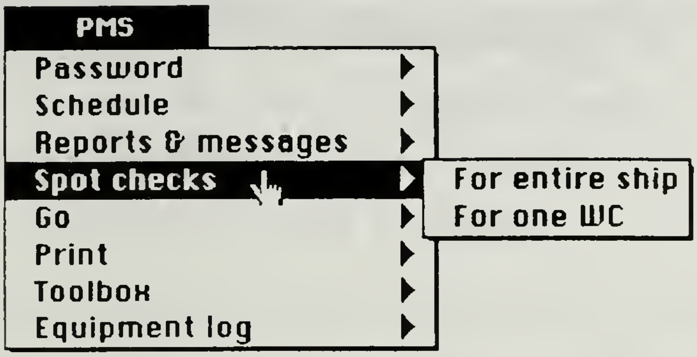


Figure 7 Spot Checks Selection

The selection, go (Figure 8), allows a user to directly look at either a MIP, MRC or Log card in the PMS system. This selection can be used provided the number of the MIP/MRC or the name of the piece of equipment is known. This selection bypasses the need of going to the appropriate database and performing a search.

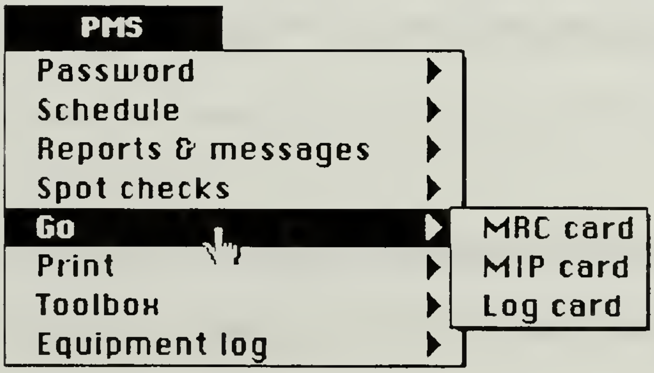


Figure 8 Go Selection

The selection, print (Figure 9), provides the user with a way of obtaining a hard copy of each of the items listed in the submenu.

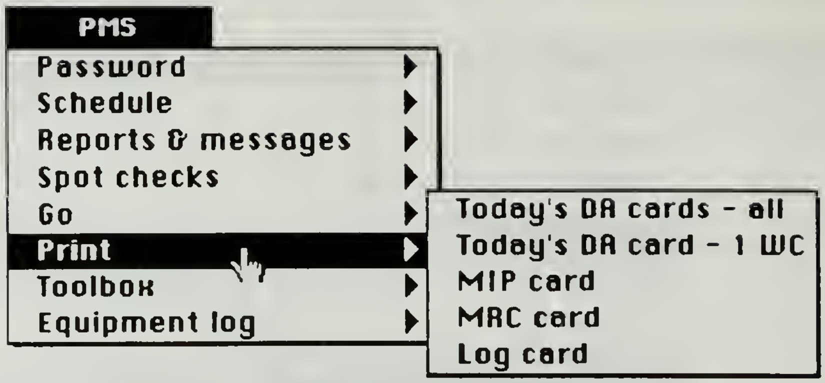


Figure 9 Print Selection

The selections, today's DA cards-all/wc, prints out a copy of the Daily Activity cards for either one or all the work centers for the current day. This function is used at the beginning of each day, so that the division officers can pass them out to their work center supervisors . Each includes a description of the jobs that the work center has to complete that day. Prior to this function being used, automatic rescheduling of uncompleted jobs takes place. This allows the system to move forward all jobs that were not completed the previous day. This function is available only to the SuperUser.

When either "MIP card", "MRC card", or "LOG card" is selected for printing, the system follows the same procedure followed for the "Go" menu item described above.

The selection, Toolbox (Figure 10), is provided only to the SuperUser of the system. It is used to perform global operations on the PMS database. These operations consist of adding new MIP cards to the database, converting MIP/MRC text files to MIP/MRC stacks, emptying the databases, making a list of all MIP/MRC cards in the database, deleting a MIP and its corresponding MRCs from the database, and for assigning the work center responsibilities for MRCs.

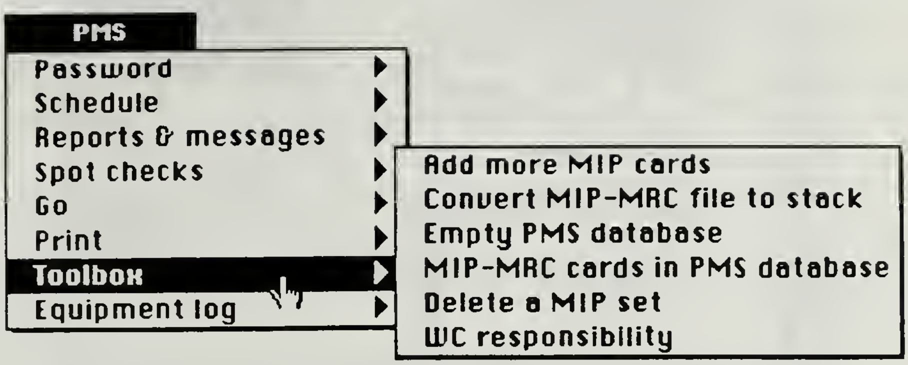


Figure 10 Tool Box Selection

The selection, equipment log (Figure 11), is available to all division officers. It provides the capability to add a new entry to the log database or remove an existing entry in cases where a piece of equipment is removed from the ship.

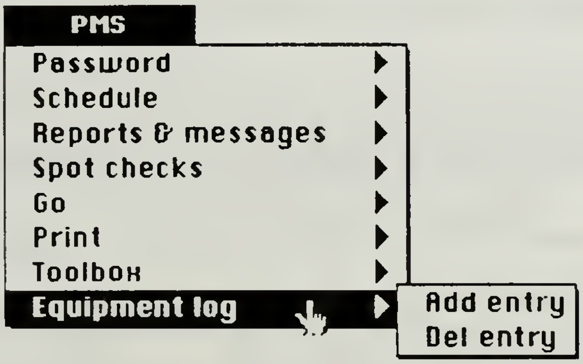
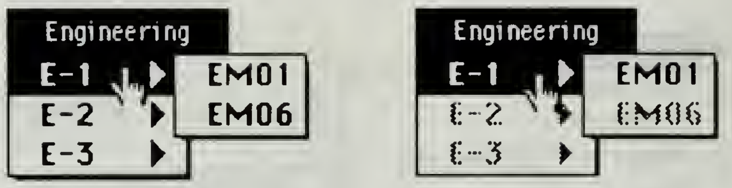


Figure 11 Equipment Log Selection

c. The division menus

There are four division menus. Each is divided into submenus for the various work centers. From these menus, the current daily activity card for their work center is shown. According to the access a user has, only the appropriate menus are activated. Samples (Figure 12) of the menus are shown below.



Engineering officer's menu EM01 personell menu

Figure 12 Division Menus

As shown in the diagram above, the Engineering officer has access to all engineering work centers. On the other hand, a member of one work center has access only to that work center. Menu items that appear dim cannot be selected.

2. The PMS Toolbox

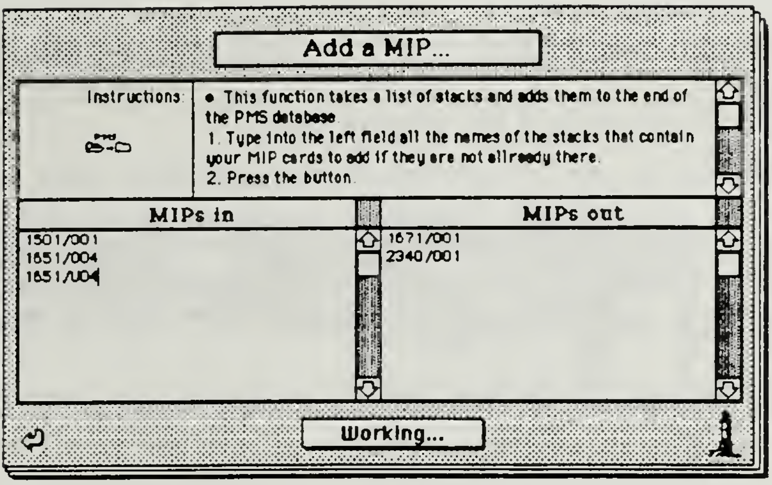
a. General description

The PMS Toolbox is a collection of utilities that allows the SuperUser to perform global manipulation of the PMS stack. Each card of this stack performs a single function. Instructions for using each function are provided on each card as help to the user.

The "Toolbox" stack presently consists of six tools. The set of tools is not static, but are the ones we thought most useful in the development of the PMS module. As the need arises, new tools can be added to the system.

b. The PMS Toolbox Tools

The "Add a MIP. . . " card (Figure 13), this function provides the capability of adding new or updated MIP cards and its corresponding MRC cards to the PMS database.

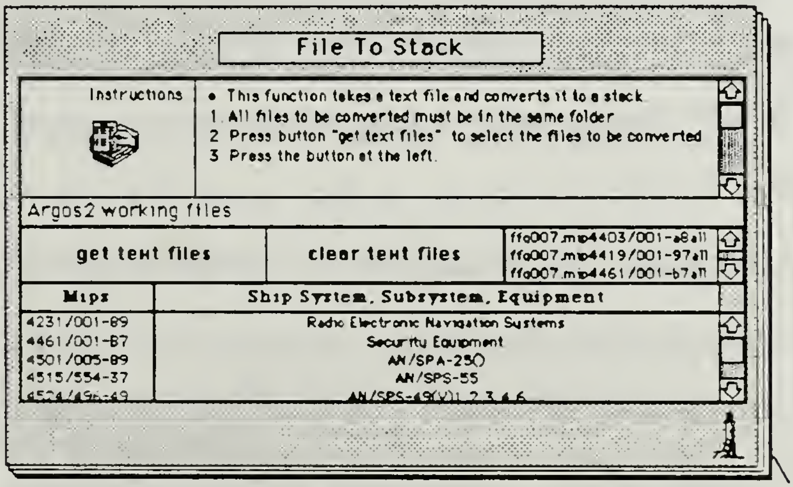


- This function takes a list of stacks and adds them to the end of the PMS database.
1. Type into the left field all the names of the stacks that contain your MIP cards to add if they are not allready there.
2. Press the button.

MIPs in	MIPs out
1501/001	1671/001
1651/004	2340/001
1651/U04	

Figure 13 Add a MIP Card

The "Convert MIP-MRC file to stack" card (Figure 14), this function provides the capability of converting PMS text files, consisting of MIPs and MRCs down loaded from the PMS database, into MIP and MRC cards.



- This function takes a text file and converts it to a stack
1. All files to be converted must be in the same folder
2. Press button "get text files" to select the files to be converted
3. Press the button at the left.

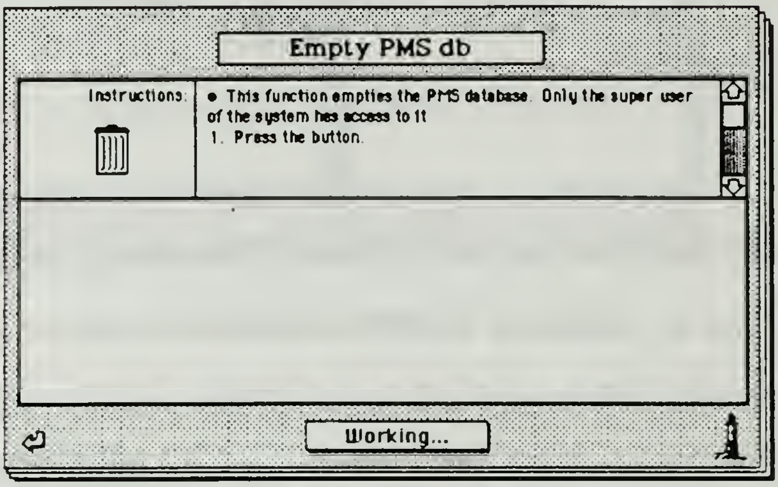
Argos2 working files

ffa007.mip4403/001-a8all
ffa007.mip4419/001-97all
ffa007.mip4461/001-b7all

Mips	Ship System, Subsystem, Equipment
4231/001-89	Radio Electronic Navigation Systems
4461/001-87	Security Equipment
4501/005-89	AN/SPA-25()
4515/554-37	AN/SPS-55
4524/496-49	AN/SPS-49(V)1,2,3,4,6

Figure 14 File to Stack Card

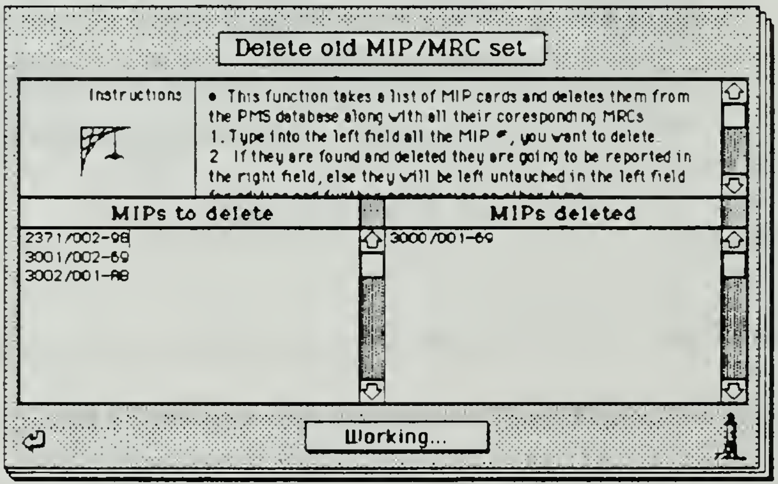
The "Empty PMS database" card (Figure 15), this function provides the SuperUser with the capability to empty the entire PMS database including all MIP and MRC cards. Six empty cards, that form the templates for the addition of new cards , are the only cards left in the database. Since this function is extremely destructive, additional check points are provided to ensure that the person requesting the execution of this function is the SuperUser.



• This function empties the PMS database. Only the super user of the system has access to it
1. Press the button.

Figure 15 Empty Pms Db Card

The "Delete a MIP set" card (Figure 16), this function provides the SuperUser with the tools to remove a MIP card along with its corresponding MRC cards. In case a MIP card is to be replaced by a newer version, this function is used to delete it. This function is set up so that even if a new version of the card is added to the system before the deletion of the old one, only the old one is deleted.



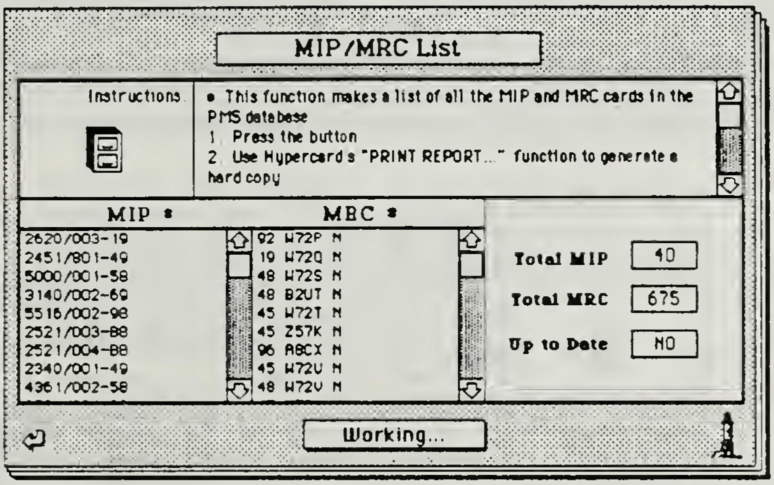
• This function takes a list of MIP cards and deletes them from the PMS database along with all their coresponding MRCs
1. Type into the left field all the MIP #, you want to delete.
2 If they are found and deleted they are going to be reported in the right field, else they will be left untauched in the left field

MIPs to delete:
2371/002-98
3001/002-69
3002/001-A8

MIPs deleted:
3000/001-69

Figure 16 Delete a MIP/MRC Card

The "MIP-MRC cards in PMS database" card (Figure 17), this function provides the user with the tool too produce a list of all the MIP and MRC numbers in the PMS database. The time of execution of this operation varies according to the size of the PMS database. To produce a printed copy of the results, the "Print Report" command of HyperCard should be used.



- This function makes a list of all the MIP and MRC cards in the PMS database
1 Press the button
2 Use Hypercard's "PRINT REPORT..." function to generate a hard copy

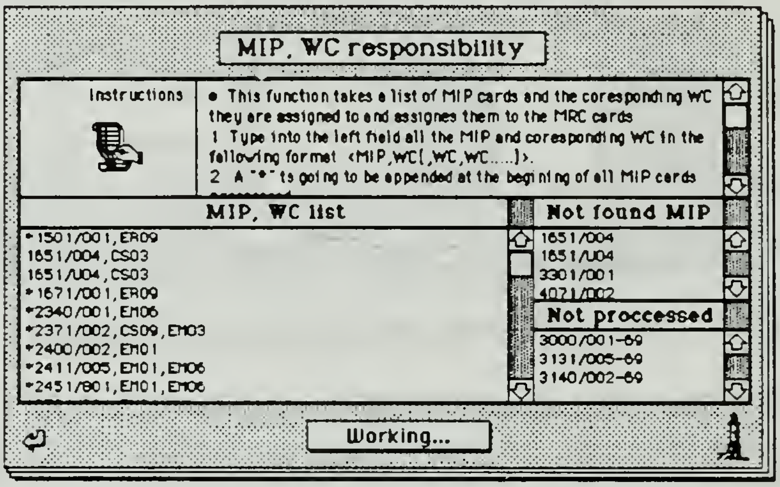
MIP *	MRC *
2620/003-19	92 W72P N
2451/801-49	19 W72Q N
5000/001-58	48 W72S N
3140/002-69	48 B2UT N
5516/002-98	45 W72T N
2521/003-88	45 Z57K N
2521/004-88	96 ABCX N
2340/001-49	45 W72U N
4361/002-58	48 W72V N

Total MIP 40
Total MRC 675
Up to Date NO

Figure 17 MIP/MRC List Card

The "WC responsibility" card (Figure 18), this function is used in the event a new set of MIP have been inserted into the PMS database. It assigns to the work centers, responsibility for the MRC cards. This information is taken into account during the automatic scheduling process. For example, if a MRC job requires two engineers and one electrician to perform it, the system schedules both the corresponding engineering work center and the electrical work center.

All MIP cards of a ship must be inserted into the "MIP,WC, responsibility" field. The format used is <MIP #,WC1,WC2,WC3. . . . . . >. If responsibility for a MIP card is modified, the SuperUser deletes the "*" from the beginning of the line the MIP is on and all the MRC's for this MIP are reprocessed. When a MIP is processed successfully, an asterisk is appended at its beginning. This controls which MIPs are going to be processed. This allows for much faster execution , because only the required or previously processed MIP cards are examined and updated. In case a MIP is not in the system, it is reported in the "Not found MIP" field . After the assignment of jobs is done, the system searches through the entire PMS database, and retrieves the MIP cards for which no work center is assigned responsibility. This provides a way of showing to the SuperUser duplicated MIPs in the PMS database.



• This function takes a list of MIP cards and the coresponding WC they are assigned to and assignes them to the MRC cards
1 Type into the left field all the MIP and coresponding WC in the following format <MIP,WC[,WC,WC.....]>.
2 A "*" is going to be appended at the begining of all MIP cards

*1501/001,ER09
1651/004,CS03
1651/U04,CS03
*1671/001,ER09
*2340/001,EM06
*2371/002,CS09,EM03
*2400/002,EM01
*2411/005,EM01,EM06
*2451/801,EM01,EM06

1651/004
1651/U04
3301/001
4071/002

3000/001-69
3131/005-69
3140/002-69

Figure 18 MIP, WC Responsibility Card

3. The "Planning" stack

a. General description

The "Planning" stack is available only to the SuperUser. This stack is used to generate and store calendars for scheduling purposes. When the SuperUser asks the system to schedule the next quarter, the system generates the calendars for the quarter. The SuperUser is then presented with the calendars to set the "ship's status" for each day of the quarter. A typical card (Figure 19) for a month is shown below.

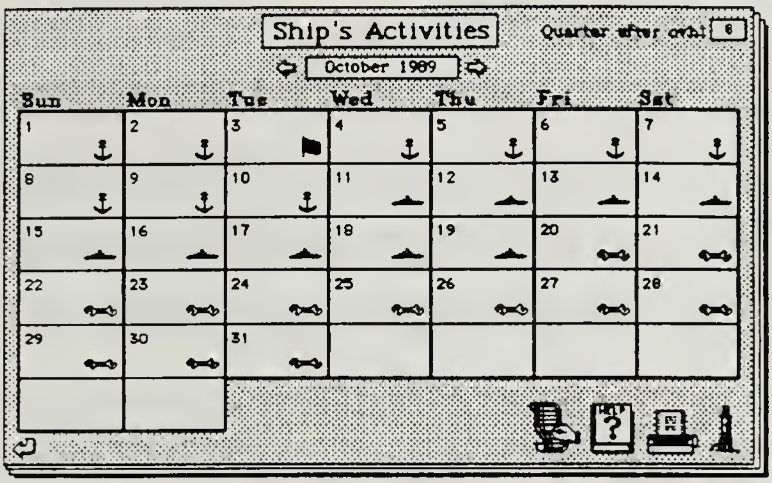


Figure 19 Ship's Activities Card

b. Scheduling Process

The actual scheduling process is initiated from this card. Once the ship's schedule has been set into the calendars for the quarter, the scheduling process can start.

Upon initiation, the system will generate a job status card for each division and a complete set of daily activities cards for the entire quarter for each division. . The job status card contains a list, numbered one to the number of MRC's, of all the MRC's that a division has responsibility for. Included in this information is the periodicity of the MRC, the total man hours involved to accomplish the MRC, any related maintenance, and the other work centers that are involved in accomplishing the completion of the job.

The periodicity of each MRC is transformed into a single letter value. The values range from "A" for a job with a periodicity of daily to a "R" if the job requires a special situation for completion. This conversion is required so that the periodicity can be used to sort the list of MRCs in descending order from R down to A. This sort order is used so that scheduling can be done on situation jobs first, with daily jobs being scheduled last.

Starting with work center EM01, the jobs are then checked one at a time for the other work center involved in completing the jobs. The jobs are then removed from these work centers so that scheduling is not duplicated. As the job is scheduled for EM01, it is simultaneously scheduled for the other cognizant work centers.

Once this is done an array consisting of the number of days of the quarter is set up. Each element of the array is contains the number of the element, the day of the week, the ship activity for that day of the week, and 0. 0 for the initial number of man hours for the day. So as to avoid scheduling major jobs on either saturday or sunday, each of these days is given an arbitrary man hours value of 1000. These days will still be scheduled for minor jobs, but not the major jobs. Once this is complete, scheduling commences with the first work center, which in this case is EM01.

The mod function plays an important role in scheduling jobs periodicity of greater than six months. As each job with a periodicity of greater than six months is schedule, it is assigned a value with a range of from two for semi annually to 20 for every five years. Then the number of the cumulative quarter is modded by this value to obtain a result. This result is checked against a set value contained in each MRC card(value indicates which quarter to schedule a job). If the two match then the job is scheduled during the current quarter. If no match, then the job will automatically be scheduled during a future quarter. At this point if the job is to be scheduled, then the array is sorted in decreasing according to the number of man hours assigned. Once sorted, then a day is picked randomly from the days which have a man-hour value of 0. 0. The ship's activity scheduled for that day is then checked to ensure that the day qualifies for assigning this particular job. If it does, then the job is scheduled for this day in all the cognizant work centers. If not, then another day is picked randomly and the process start all over until either the job is scheduled or ten day have been picked at random and all have failed. If the job is unscheduled at this point, then the program starts at the first element of the array and searches down until a day is found where the job can be scheduled.

Jobs with a periodicity of less than six months are handled in the following manners. Quarterly jobs are handled using the same random selection process as described above. The only difference is that since quarterly jobs are always done each quarter, they are not put through the same quarter selection process as described above. For two week, and less periodicity jobs, the array is resorted back so that the days of the quarter are back in correct order.

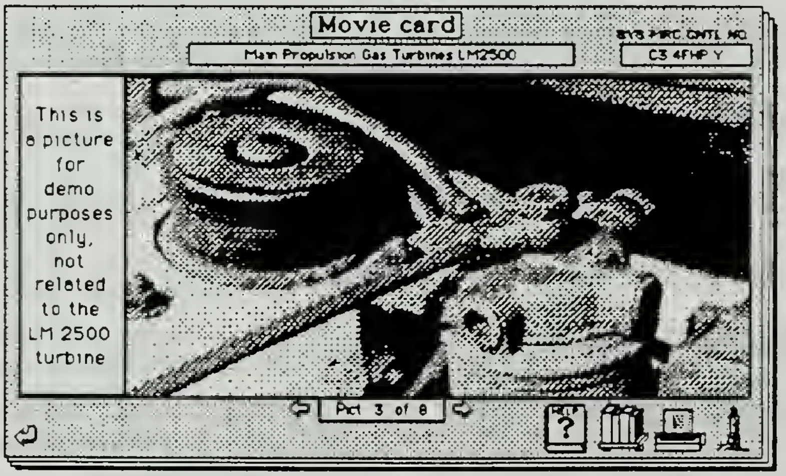
For two week jobs, the array is divided by 14 to produce the number of two week periods during the quarter. Then the individual two week periods, taken one at a time, are sorted in a separate variable according to the total man hours each day has. The selection process then starts with the day with the fewest number of man hours looking for a day on which the job can be scheduled. If no day is appropriate then the job is reported as being unscheduled during this two week period. The same process is taken for weekly jobs except the periods are only seven days long. Jobs with three day, two day, and daily periodicities are scheduled accordingly.

One final word on the scheduling process. As jobs are scheduled, the system attempts to maintain an even distribution of total man hours through out the quarter. To do this, the system attempts to schedule jobs on days that have the fewest total man hours.

4. The "Movies" stack

a. General description

The "Movies" stack is a collection of still pictures taken with an ordinary camera and scanned into the system. It is used for demonstration purposes to give the user a better feel of the maintenance to be completed, and to point out intricate parts of an actual PMS job. Still pictures can be used to show equipment parts that are either difficult to describe and/or jobs that are difficult to complete without on the spot supervision. The role this function will play is in the training of the personnel. The ability to see a job before it is to be actually performed should help individuals in performing the maintenance jobs, unattended. Additionally, pictures will allow an individual to review a particular job at their own pace and give an instructor the tools to better explain a job before it is performed. A sample picture for a fictious MRC card (Figure 20) is shown below.



Main Propulsion Gas Turbines LM2500

SYS MRC CNTL NO: C3 4FHP Y

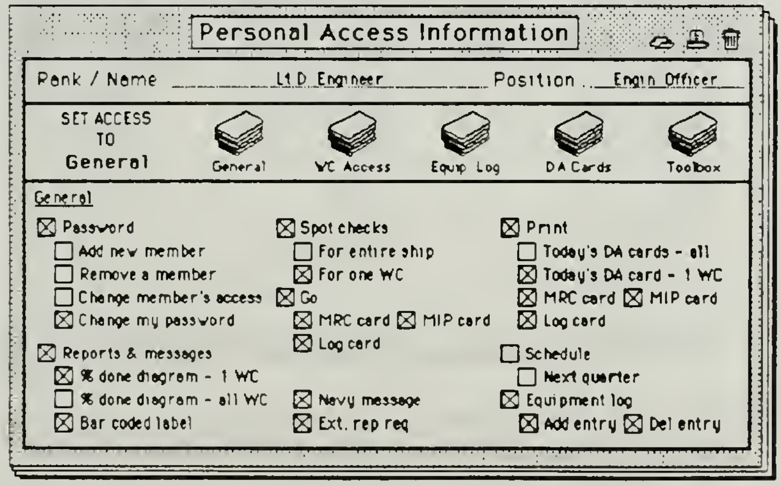
This is a picture for demo purposes only, not related to the LM 2500 turbine

Figure 20 Movie Card

5. The "Password" stack

a. General description

The "Password" stack is the heart of the safeguard system for PMS module. The stack contains one card (Figure 21) for each person, with access to the system, and contains the information on what they have access to. This is how the system restricts users from accessing functions they are not authorized to use. If an unauthorized user tries to open this stack and review or change its contents, the program automatically forces them out the PMS system. The password system's card is divided into five individual sections. Selecting one of the five stack buttons at the top, provides access to the various sections of the password system. The SuperUser selects the items that the person whose card he is working on is to have access to. Each box corresponds to an item in the various menus of the PMS system. After the desired items have been set, the SuperUser can store, print or delete the current card by pressing one of the respective buttons at the top of the card. The very first time the PMS system is used, a card, named "installer", is provided to allow full access to the system by the person who is installing the system. This password should be changed immediately by the SuperUser.



Rank / Name Lt D. Engineer
Position Engin Officer

General

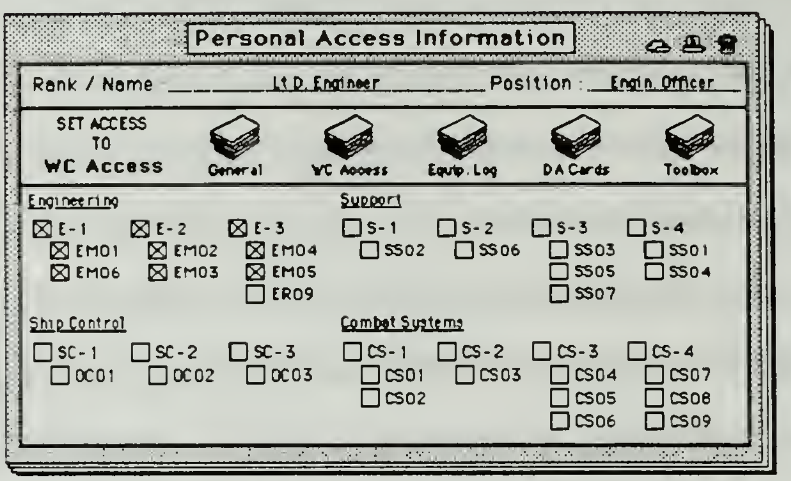
☒ Password
  ☐ Add new member
  ☐ Remove a member
  ☐ Change member's access
  ☒ Change my password
☒ Reports & messages
  ☒ % done diagram - 1 WC
  ☐ % done diagram - all WC
  ☒ Bar coded label

☒ Spot checks
  ☐ For entire ship
  ☒ For one WC
☒ Go
  ☒ MRC card ☒ MIP card
  ☒ Log card
  ☒ Navy message
  ☒ Ext. rep req

☒ Print
  ☐ Today's DA cards - all
  ☒ Today's DA card - 1 WC
  ☒ MRC card ☒ MIP card
  ☒ Log card
☐ Schedule
  ☐ Next quarter
☒ Equipment log
  ☒ Add entry ☒ Del entry

Figure 21 Personal Access Info Card, General

The WC access button presents to the SuperUser, the work center menus (Figure 22). It is used to control access to the DA cards for the various work centers.



Rank / Name: Lt D. Engineer
Position: Engin. Officer

Engineering
- ☒ E-1
  - ☒ EM01
  - ☒ EM06
- ☒ E-2
  - ☒ EM02
  - ☒ EM03
- ☒ E-3
  - ☒ EM04
  - ☒ EM05
  - ☐ ER09

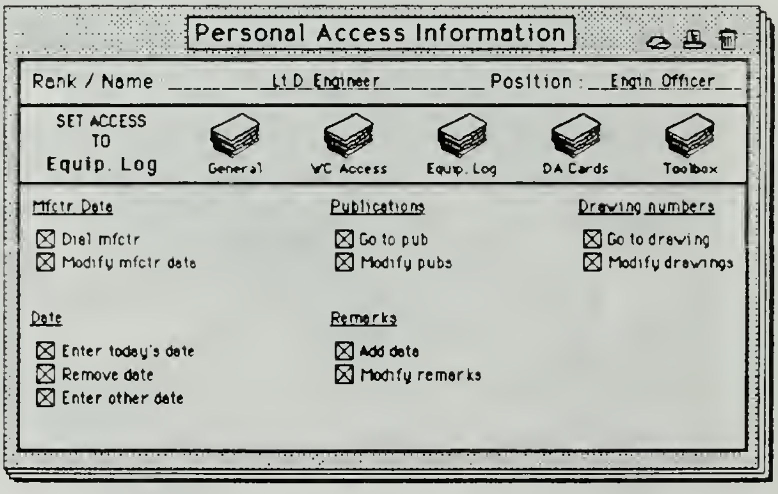
Support
- ☐ S-1
  - ☐ SS02
- ☐ S-2
  - ☐ SS06
- ☐ S-3
  - ☐ SS03
  - ☐ SS05
  - ☐ SS07
- ☐ S-4
  - ☐ SS01
  - ☐ SS04

Ship Control
- ☐ SC-1
  - ☐ OC01
- ☐ SC-2
  - ☐ OC02
- ☐ SC-3
  - ☐ OC03

Combat Systems
- ☐ CS-1
  - ☐ CS01
  - ☐ CS02
- ☐ CS-2
  - ☐ CS03
- ☐ CS-3
  - ☐ CS04
  - ☐ CS05
  - ☐ CS06
- ☐ CS-4
  - ☐ CS07
  - ☐ CS08
  - ☐ CS09

Figure 22 Personal Access Info Card, WC

The equipment log button presents to the SuperUser the options (Figure 23) available in the Equipment log stack. In general, access to these functions is given only to division officers and higher.



Rank / Name: Lt D Engineer
Position: Engin Officer

Mfctr Data
- ☒ Dial mfctr
- ☒ Modify mfctr data

Publications
- ☒ Go to pub
- ☒ Modify pubs

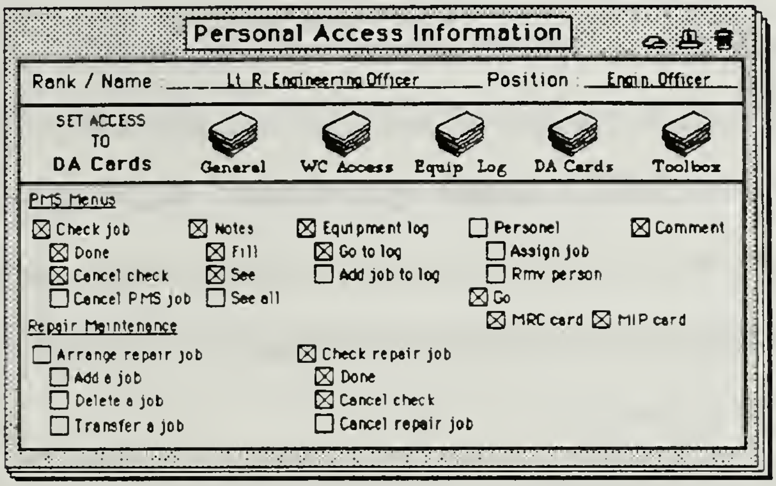
Drawing numbers
- ☒ Go to drawing
- ☒ Modify drawings

Date
- ☒ Enter today's date
- ☒ Remove date
- ☒ Enter other date

Remarks
- ☒ Add data
- ☒ Modify remarks

Figure 23 Personal Access Info Card, Equip

The DA cards button presents to the SuperUser the options (Figure 24) for the Daily Activity cards. It is used to control access to the daily activities cards. Generally, everyone on board ship is given access to these cards. According to the different options selected, different menus will be available to them.



Rank / Name : Lt R. Engineering Officer
Position : Engin. Officer

PMS Menus

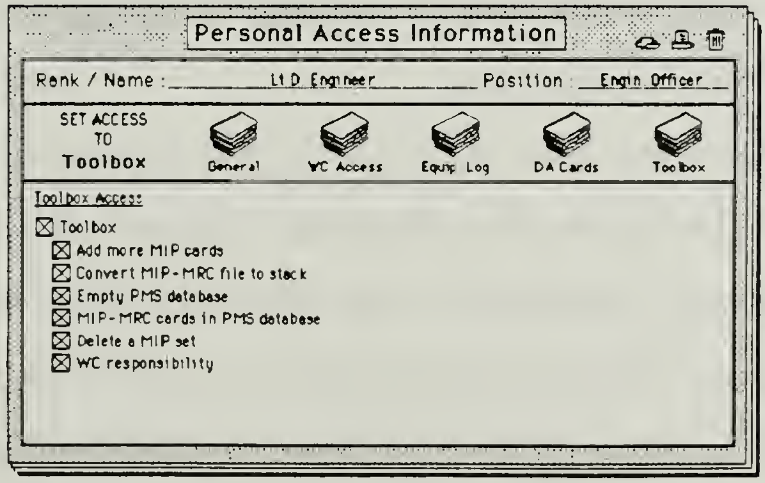
- ☒ Check job
  - ☒ Done
  - ☒ Cancel check
  - ☐ Cancel PMS job
- ☒ Notes
  - ☒ Fill
  - ☒ See
  - ☐ See all
- ☒ Equipment log
  - ☒ Go to log
  - ☐ Add job to log
- ☐ Personel
  - ☐ Assign job
  - ☐ Rmv person
- ☒ Go
  - ☒ MRC card ☒ MIP card
- ☒ Comment

Repair Maintenance

- ☐ Arrange repair job
  - ☐ Add a job
  - ☐ Delete a job
  - ☐ Transfer a job
- ☒ Check repair job
  - ☒ Done
  - ☒ Cancel check
  - ☐ Cancel repair job

Figure 24 Personal Access Info Card, Da

The toolbox button presents to the SuperUser, the Toolbox access menu (Figure 25). It is used to control access to the Toolbox menu of the driver. In general , only the SuperUser has access to these functions.



Rank / Name : Lt D. Engineer
Position : Engin. Officer

Toolbox Access:

- ☒ Toolbox
  - ☒ Add more MIP cards
  - ☒ Convert MIP-MRC file to stack
  - ☒ Empty PMS database
  - ☒ MIP-MRC cards in PMS database
  - ☒ Delete a MIP set
  - ☒ WC responsibility

Figure 25 Personal Access Info Card, Toolbox

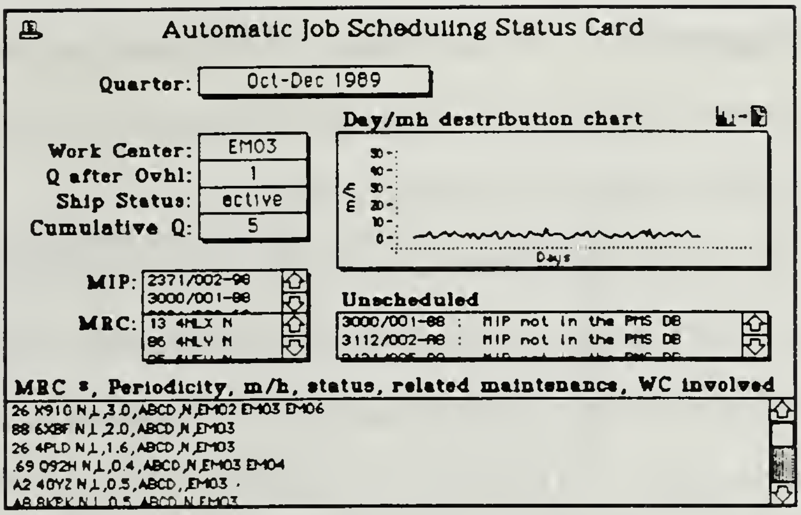
6. The "Daily Activity" stack

If the PMS driver stack can be compared to the bridge of a ship, the Daily activity stack is the engineering control room. This stack contains cards that are used in the day to day operation of the PMS module as jobs are scheduled and then assigned for completion.

A daily activity stack is automatically generated every time a new quarter is scheduled. The stack consists of two types of cards. The first ,the "Automatic job Scheduling Status Card", contains information about the job scheduling done for this quarter. The second, the actual Daily activity cards, stores information about the jobs that are scheduled on each day of a quarter. In depth description of each of these cards is given below.

a. The "Automatic Job Scheduling Status Card"

For every work center, one card (Figure 26) of this type is generated and set at the beginning of each quarter. This card exists to help the division officer evaluate the system, showing information that can be used if corrective action needs be taken to improve the overall performance of the PMS system. A graph provides a visual representation showing the distribution of the jobs throughout the quarter. Jobs that were not scheduled are also reported, along with the reason that automatic scheduling has failed on these jobs.


Automatic Job Scheduling Status Card

Quarter: Oct-Dec 1989

Work Center: EM03
Q after Ovhl: 1
Ship Status: active
Cumulative Q: 5

MIP: 2371/002-98
3000/001-88

MRC: 13 4HLX N
86 4HLY N

Unscheduled
3000/001-88 : MIP not in the PMS DB
3112/002-A8 : MIP not in the PMS DB

MRC #, Periodicity, m/h, status, related maintenance, WC involved
26 X910 N⊥,3.0,ABCD,N,EM02 EM03 EM06
88 6XBF N⊥,2.0,ABCD,N,EM03
26 4PLD N⊥,1.6,ABCD,N,EM03
.69 092H N⊥,0.4,ABCD,N,EM03 EM04
A2 40YZ N⊥,0.5,ABCD,,EM03

Figure 26 Scheduling Status Card

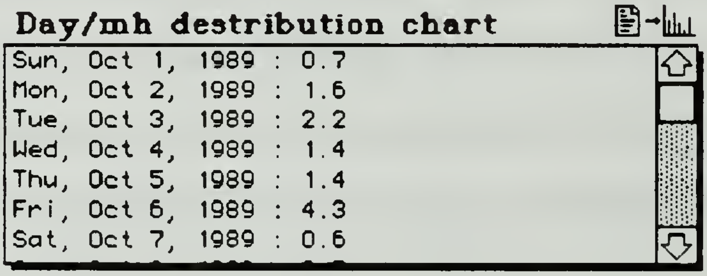
Most of the fields of the scheduling status card are self explanatory but a few need explaining.

The cumulative quarter is a number that shows the number of quarters since the PMS module was installed. The purpose of this number is for scheduling jobs that do not belong in any particular quarter, such as six-month and annual jobs. Using the mod function, the system decides which jobs, with a periodicity greater than "quarterly", will be scheduled during a quarter and which will be skipped. This way the system totally eliminates the need of "long term" scheduling because jobs with a periodicity greater than quarterly will automatically scheduled during the desired quarter.

The unscheduled field is one of the most important for the whole system, because it provides feedback information to both the PMS coordinator and the division officer. It shows which jobs were automatically scheduled along with the reason the system failed to do so. Some jobs, like "situation requirement" jobs, cannot be scheduled automatically and the division officer will have to schedule them manually. In this field, MIP and MRC cards not in the PMS database are reported so that the PMS coordinator can add them. The system also reports jobs, with a periodicity longer than "quarterly", if the desired quarter to be scheduled is not set. This is set in the "data" field of the MRC cards.

The "MRC#, Periodicity, m/h, status,related maintenance,WC involved" field. After searching the PMS database to determine whether or not a MRC cards exists,the system places, into this field, information relevant to each MRC card to be schedule. The system then deletes this job from the other work centers involved, so that the same job is scheduled on the same day for all the work centers. After setting this field for all work centers, the system then takes the MRC jobs, one by one, and starts the scheduling process.

The "Day/mh distribution chart" field. The information in this field is shown in two formats . The first is the list view, where the user is presented with a digital readout of the man/hours scheduled. An example of this view (Figure 27) is shown below.



Day/mh destribution chart

```
Sun, Oct 1, 1989 : 0.7
Mon, Oct 2, 1989 : 1.6
Tue, Oct 3, 1989 : 2.2
Wed, Oct 4, 1989 : 1.4
Thu, Oct 5, 1989 : 1.4
Fri, Oct 6, 1989 : 4.3
Sat, Oct 7, 1989 : 0.6
```

Figure 27 DAY/MH Distribution Field, List View

The second format is the graph view. The user is provided with an analog readout presenting an immediate view of whether or not the scheduling has been done efficiently. The smoother the curve, the smaller the peaks (the difference between minimum and maximum in the m/h scale), the more efficiently the scheduling has been done. The graphic form (Figure 28) is shown below.

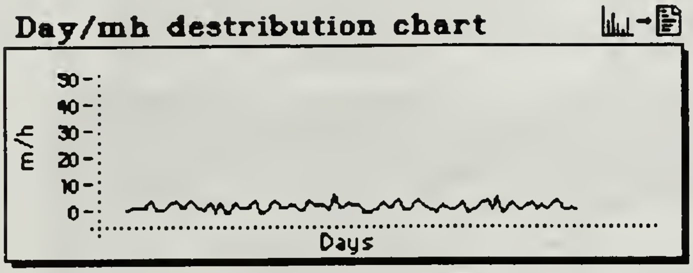


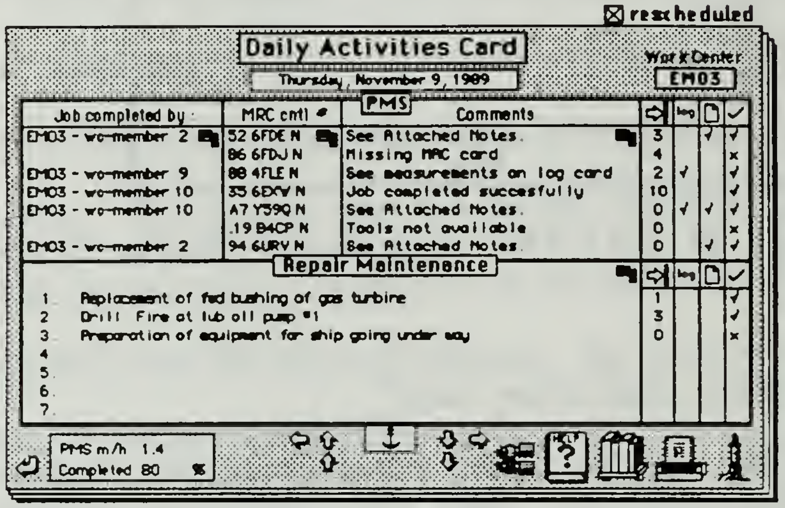
Figure 28 DAY/MH Distribution Field, Graph View

7. The "Daily Activities Card"

The "Daily Activity" cards are located after the "Automatic Job Scheduling Status" cards in the Daily Activity stack. One card is generated for each day of the quarter for each work center. This card has two purposes. The first as an assignment sheet for each work center, where the jobs, both for preventive and repair, are found. The second as a database for future reference and report generation.

The user can view directly the DA card of today from the work center menus of the PMS driver. The first time the card is opened each day, automatic rescheduling of uncompleted jobs from the previous day takes place. In a typical day, the person responsible for assigning the PMS jobs, assigns individuals to them using the "Personnel" function that is described below. The division officer and the head of the work center, using the PMS man/hours for the day as a reference, can assign repair maintenance jobs to be done for the particular day. In case they feel that a job can wait until the next day, they can leave the job unassigned and the job will automatically be transferred to the next day.

The card (Figure 29) is divided to two sections. The top section is for recording Preventive maintenance and the bottom section is for recording repair maintenance.



☒ rescheduled

Daily Activities Card

Thursday, November 9, 1989

Work Center
EM03

PMS

Job completed by :	MRC cntl #	Comments	⇨	log	🗋	✓
EM03 - wc-member 2	52 6FDE N	See Attached Notes.	3		✓	✓
	86 6FDJ N	Missing MRC card	4			×
EM03 - wc-member 9	88 4FLE N	See measurements on log card	2	✓		✓
EM03 - wc-member 10	35 6EXV N	Job completed succesfully	10			✓
EM03 - wc-member 10	A7 Y59Q N	See Attached Notes.	0	✓	✓	✓
	.19 B4CP N	Tools not available	0			×
EM03 - wc-member 2	94 6URV N	See Attached Notes.	0		✓	✓

Repair Maintenance

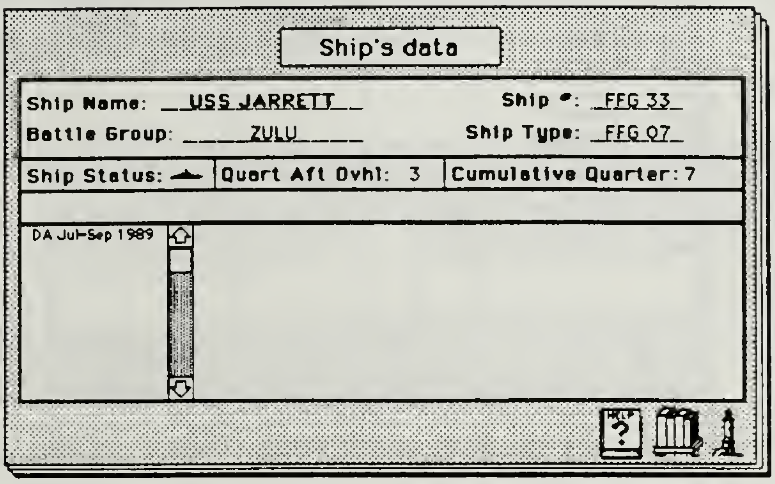
		⇨	log	🗋	✓
1.	Replacement of fed bushing of gas turbine	1			✓
2	Drill Fire at lub oil pump #1	3			✓
3	Preparation of equipment for ship going under way	0			×
4					
5.					
6.					
7.					

PMS m/h 1.4
Completed 80 %

Figure 29 Daily Activities Card

8. The "Ship Data" stack

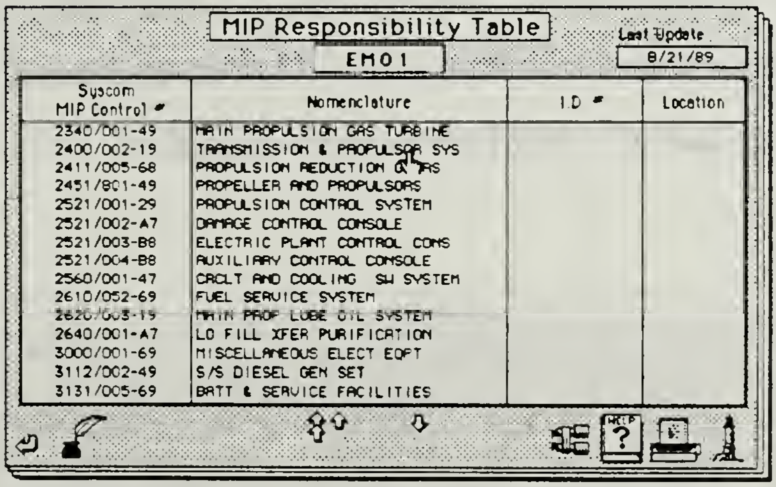
The ship data stack consists of three types of cards, the ships data card (Figure 30), the mip responsibility table (Figure 31), and the personnel/work center (Figure 32) cards. The ships data card contains such information as whether or not the ship is in active status, the number of quarters after its last overhaul, and the number of quarters since the PMS module was installed on board ship. The mip responsibility table contains a card for each work center. On it is list of the MIPs that the work center is responsible for. The personnel/work center contains a card for each work center with a list of the individuals that are assigned to that work center. Information contained in these cards is used in the automatic scheduling process and for assigning personnel to jobs once the jobs are scheduled.



Ship Name: USS JARRETT
Ship #: FFG 33
Battle Group: ZULU
Ship Type: FFG 07
Ship Status:
Quart Aft Ovhl: 3
Cumulative Quarter: 7

DA Jul-Sep 1989

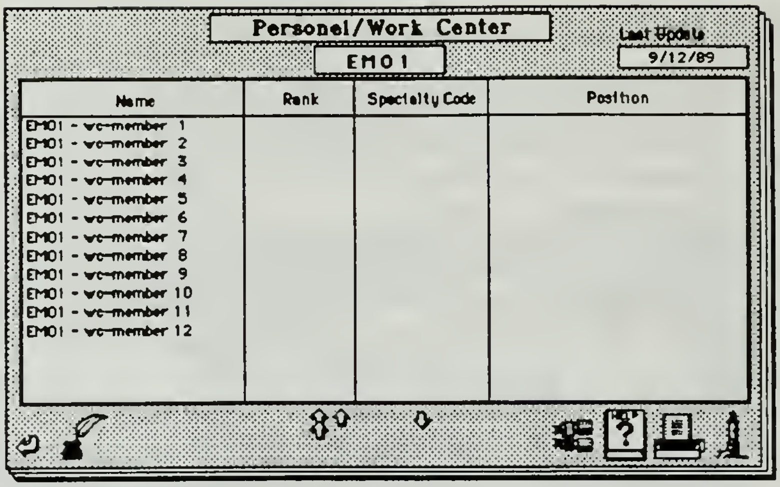
Figure 30 Ship's Data Card



Last Update 8/21/89

Syscom MIP Control #	Nomenclature	I.D #	Location
2340/001-49	MAIN PROPULSION GAS TURBINE		
2400/002-19	TRANSMISSION & PROPULSOR SYS		
2411/005-68	PROPULSION REDUCTION GRS		
2451/801-49	PROPELLER AND PROPULSORS		
2521/001-29	PROPULSION CONTROL SYSTEM		
2521/002-A7	DAMAGE CONTROL CONSOLE		
2521/003-B8	ELECTRIC PLANT CONTROL CONS		
2521/004-B8	AUXILIARY CONTROL CONSOLE		
2560/001-47	CRCLT AND COOLING SW SYSTEM		
2610/052-69	FUEL SERVICE SYSTEM		
2620/003-19	MAIN PROP LUBE OIL SYSTEM		
2640/001-A7	LO FILL XFER PURIFICATION		
3000/001-69	MISCELLANEOUS ELECT EQPT		
3112/002-49	S/S DIESEL GEN SET		
3131/005-69	BATT & SERVICE FACILITIES		

Figure 31 MIP Responsibility Card



Personel/Work Center

EM01

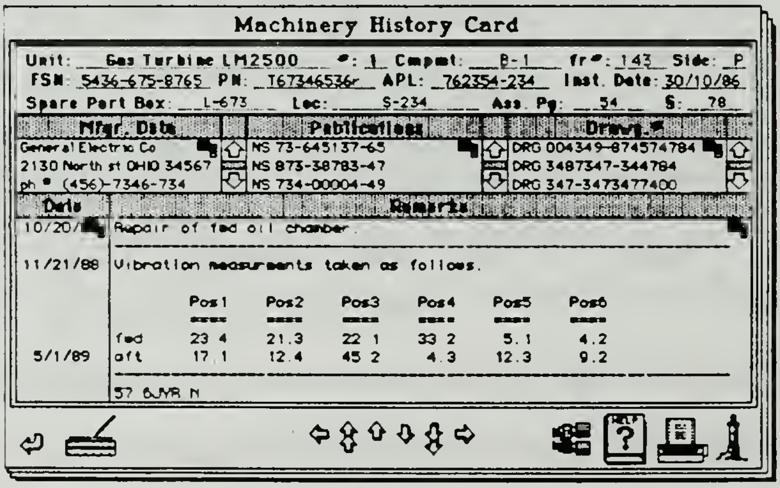
Last Update: 9/12/89

Name	Rank	Specialty Code	Position
EM01 - wc-member 1			
EM01 - wc-member 2			
EM01 - wc-member 3			
EM01 - wc-member 4			
EM01 - wc-member 5			
EM01 - wc-member 6			
EM01 - wc-member 7			
EM01 - wc-member 8			
EM01 - wc-member 9			
EM01 - wc-member 10			
EM01 - wc-member 11			
EM01 - wc-member 12			

Figure 32 Personnel Work Center Card

9. The "Log" stack

The log stack consists of machinery history cards (Figure 33), one for each piece of equipment carried on board ship, used in tracking maintenance. Each card contains the complete maintenance history of the equipment, a complete list of technical manual numbers for reference use, a list of applicable drawing numbers, and data on the manufacturer if outside help is required.



Machinery History Card

Unit: Gas Turbine LM2500 #: 1 Cmpmt: B-1 fr#: 143 Side: P
FSN: 5436-675-8765 PN: T67346536r APL: 762354-234 Inst. Date: 30/10/86
Spare Part Box: L-673 Loc: S-234 Ass. Pg: 54 §: 78

Mfgr. Data	Publications	Drwg. #
General Electric Co	NS 73-645137-65	DRG 004349-874574784
2130 North st OHIO 34567	NS 873-38783-47	DRG 3487347-344784
ph # (456)-7346-734	NS 734-00004-49	DRG 347-3473477400

Date	Remarks
10/20/	Repair of fwd oil chamber.
11/21/88	Vibration measurements taken as follows.

	Pos1	Pos2	Pos3	Pos4	Pos5	Pos6
fwd	23 4	21.3	22 1	33 2	5.1	4.2
aft	17 1	12.4	45 2	4.3	12.3	9.2

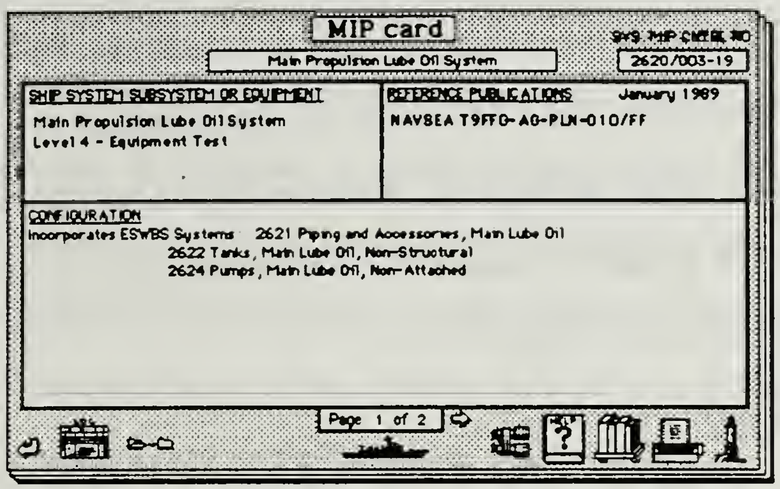
5/1/89

57 6JYR N

Figure 33 Machinery History Card

10. The "PMS" stack

The PMS stack consists of the entire database of MIPs (Figure 34) and the associated MRCs (Figure 35) for an entire ship.



SYS MIP CNTRL NO: 2620/003-19

Main Propulsion Lube Oil System

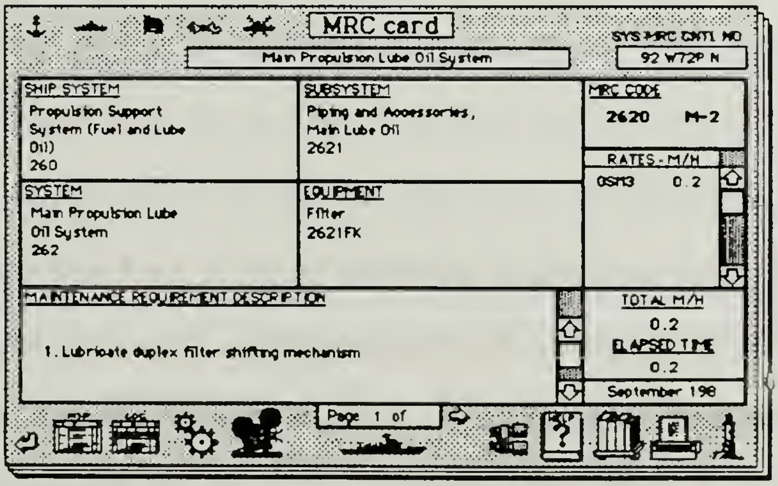
SHIP SYSTEM SUBSYSTEM OR EQUIPMENT
Main Propulsion Lube Oil System
Level 4 - Equipment Test

REFERENCE PUBLICATIONS January 1989
NAVSEA T9FFG-AG-PLN-010/FF

CONFIGURATION
Incorporates ESWBS Systems 2621 Piping and Accessories, Main Lube Oil
2622 Tanks, Main Lube Oil, Non-Structural
2624 Pumps, Main Lube Oil, Non-Attached

Figure 34 MIP Card

The original architects of Argos concentrated their efforts on the FFG-7 class of ships in designing their prototype. The decision was made to continue their effort by implementing the PMS stack so that it would be relevant to that class of ships. As such, the PMS stack is a complete MIP/MRC database for the FFG-7 class of ships.



SYS MRC CNTL NO: 92 W72P N

Main Propulsion Lube Oil System

SHIP SYSTEM
Propulsion Support System (Fuel and Lube Oil)
260

SUBSYSTEM
Piping and Accessories, Main Lube Oil
2621

MRC CODE
2620 M-2

SYSTEM
Main Propulsion Lube Oil System
262

EQUIPMENT
Filter
2621FK

RATES - M/H
OSM3 0.2

MAINTENANCE REQUIREMENT DESCRIPTION
1. Lubricate duplex filter shifting mechanism

TOTAL M/H
0.2

ELAPSED TIME
0.2

September 198

Figure 35 MRC Card

Using the Navy's on database, individual PMS stacks can be prepared for each class of ship in service. Having been down-loaded directly from the Navy's own PMS database, the information contained in the PMS stack is as up to date as possible.

Every attempt was made to make accessing the PMS stack as user friendly as possible. Therefore, the PMS stack can be accessed sequentially or randomly depending on the needs of the user. Random access is is provided through the built in search function located on each card and by directly selecting an individual MRC number on each of the MIP cards.

E. SUMMARY

Although the PMS module as implemented is fully functional and ready to use, it is not all inclusive. Tthe modularity of the PMS and ARGOS modules allows functions to be added and deleted as necessary. The PMS module demonstrates the power and capabilities of ARGOS and as more and more users interact with the module new methods and options will be discovered and implemented.

V. CONCLUSIONS

The purpose of Argos is to enhance personnel productivity through an emphasis on human engineering, reducing paperwork, increasing accessibility of systems, and expanding the use of decision systems. With this in mind, our goal for this thesis was to attempt to construct a computerized system that would meet the present and future needs of the Navy in the area of planned maintenance. We wanted to provide a completely self contained system capable of performing all the required tasks that are found in the current system, but do it in a way that would make the system run more smoothly and efficiently.

When we started the design process, we were not completely sure that all of the tasks were possible to perform using a computer. Specifically, we were apprehensive about whether or not an automatic scheduling system could in fact be implemented. In our initial attempts at implementing the process we were unsuccessful in the scheduling of of jobs with a periodicity of greater than quarterly. However through many different attempts, we finally came up with a unique solution, which we detailed in chapter four of this thesis.

We were also unsure as how to develop our database of MIP and MRC cards. In our experience we found that the on-line data that we had to work with, which was in a standard format, at times would deviate from the standard format. The possible deviations produced a lot of inconsistencies in the data that we had to work with and caused us problems in our attempts to use this data for conversion to the format we wished to use for our implementation. We were able to develop a scheme that takes these inconsistencies into account. The process as developed ensures that using the on-line database, we will have a system that will allow for easy addition of information to the database and which will be in a consistent format.

Graphics are a large part of the information for our database, however we were unable to fully complete the conversion of all graphics from hard copy format to our format. Complete conversion of all required graphics, done easily through the use of a Macintosh compatible scanner, will be left for future implementation.

Testing of our Implementation has been conducted in the laboratory with very successful results. However we believe that our module must be fully tested under live test conditions and circumstances by the people for which it was intended before final judgement can be made.

To make the PMS module useful in its intended format, we believe that the system should be set up on a local area network with remote access terminals distributed throughout the ship.

We believe we were successful in our goal of designing such a system and that the modules we implemented are the first real improvement in the system as it is now implemented . We were not only able to duplicate all the required tasks but to increase the efficiency and performance of each of these tasks.

Through Hypercard, we were able to design a system that should show a reduction in the paperwork load. We believe that if the PMS module is used in the fleet, it would offer considerable benefits and savings to the Navy.

The Maintenance area of ARGOS has many areas that still need to be analyzed and designed. Continuing research on the project would be most advantageous for the Navy.

Some topics for further research are listed below:

- Develop the PMS module using the object oriented programming language on the NeXT ™ computer.
- Develop the PMS module on a Macintosh system, such as SuperCard™, that would allow for both expanded size windows and multiple screens.
- Implement the complete set of graphics for the PMS cards as well as for the Movies section of the PMS module.

APPENDIX A

Argos

PMS MODULE

User's Manual

C. T. Wu
Lt. H. V. Turner, USN
Lt. D. Antonopoulos, Hellenic Navy

Naval Postgraduate School, Monterey CA
December 1989

TABLE OF CONTENTS

INSTALLATION OF THE SYSTEM **54**
- HARDWARE REQUIREMENTS 54
- INSTALLING THE PMS MODULE 54

THE PMS DRIVER **57**
- GENERAL DESCRIPTION 57
- LOGGING IN THE PMS SYSTEM 57
- THE MENUS 59
  - The PMS menus 59
  - The Division menus 73

THE PMS TOOLBOX **74**
- GENERAL DESCRIPTION 74
- THE PMS TOOLBOX TOOLS 74
  - "Add a MIP..." 74
  - "Convert MIP-MRC file to stack" 75
  - "Empty PMS database" 77
  - "Delete a MIP set" 78
  - "MIP-MRC cards in PMS database" 79
  - "WC responsibility" 80

THE "PLANNING" STACK **83**
- GENERAL DESCRIPTION 83

THE "MOVIES" STACK **88**
- GENERAL DESCRIPTION 88

THE "PASSWORD" STACK **90**
- GENERAL DESCRIPTION 90

THE DAILY ACTIVITY STACK **95**
- "AUTOMATIC JOB SCHEDULING STATUS CARD" 95
- THE "DAILY ACTIVITIES CARD" 99

The Preventive Maintenance (top part)....................102

The menus....................104

The Repair or Corrective Maintenance (bottom part)....................109

The menu....................110

THE SHIP DATA STACK....................113

GENERAL DESCRIPTION....................113

THE "SHIP'S DATA" CARD....................113

THE "MIP RESPONSIBILITY TABLE" CARD....................113

THE "PERSONNEL/WORK CENTER" CARD....................114

THE LOG STACK....................116

GENERAL DESCRIPTION....................116

THE REPORTS STACK....................125

GENERAL DESCRIPTION....................125

THE PMS STACK....................127

GENERAL DESCRIPTION....................127

THE "MIP/MRC LIST" CARD....................127

THE "MIP" CARD....................128

MIP 1st page....................128

MIP 2nd page....................131

THE MRC CARD....................132

MRC 1st page....................132

MRC 2nd page....................138

MRC 3rd page....................139

MRC 4th page....................139

THE AUXILIARY STACKS....................140

GENERAL DESCRIPTION....................140

THE "RESOURCE STORING" STACKS....................140

The "Home" stack....................140

The "Templates" stack....................140

The "General Label Storage" stack....................141

THE "PRINTING" STACKS....................141

The "Log_report" stack..141
The "DA_report" stack..142
The "Generic_fld_print" stack..143
The "MRC_print" stack..143
The "MIP_printed" stack..145
The "Compl_report" stack..145
The "Spot Check" stack..145

Installation of the system

Hardware Requirements

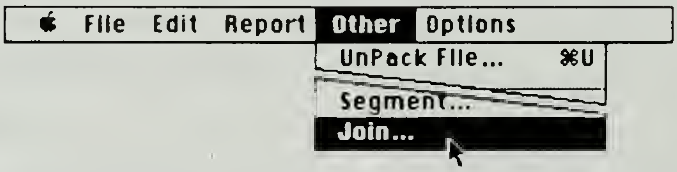
To use the PMS module of Argos you need the following:

- A Macintosh computer with at least 1 MB (preferably 2 MB) of memory.
- One double-sided (800K) drive.
- A hard disk drive with 30 MB or more free space.
- "Stuffit 1.5.1", a public domain utility to be used during the installation process.
- HyperCard version 1.2.1 or later.
- The original Argos module, if you want the link to equipment capability to be available.

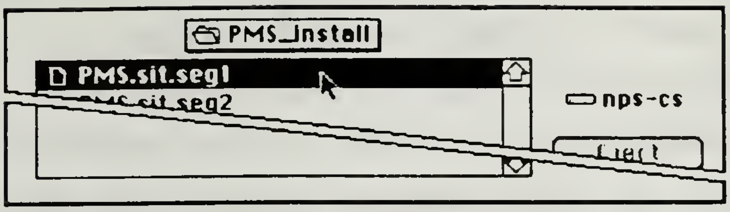
Installing the PMS module

- Turn on your Macintosh.
- Create a new folder and name it "PMS_install".
- Insert the disk called "PMS-Installation Utilities" and copy the file "Stuffit" into this folder.

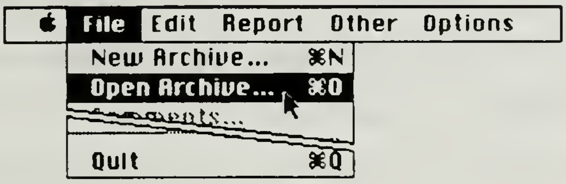
- Insert one by one the rest of the disks ("PMS.sit.seg1"..."PMS.sit.seg10") and copy the single file that they contain into the same folder ("PMS_install").
- When this process is over, double click on the icon of "Stuffit" in order to start the application.
- Select the item <Join...> from the <Other> menu.



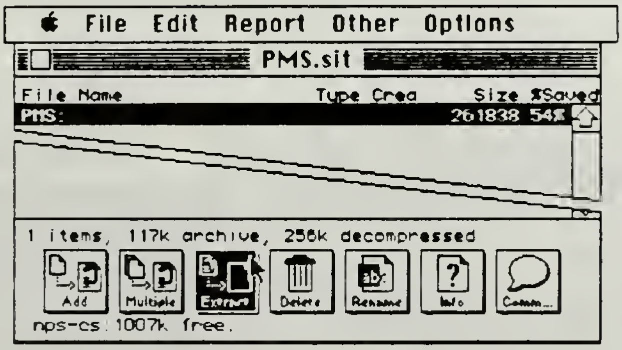
- When the dialog box appears, select all the segments (PMS.sit.seg1...) one by one.



- After all the segments have been joined, quit the "Stuffit" application. An icon with the name "PMS.sit" will be stored in the "PMS_install" folder. Delete all the segment files named "PMS.sit.seg1"..."PMS.sit.seg10", in order to conserve space on the hard disk drive.
- Restart the "Stuffit" program, and select <Open Archive...> from the file menu.



- A dialog box will appear. Double click on the file "PMS.sit".
- The screen shown below, will appear. Select the file PMS: and click "Extract".



- On the next dialog select "Save" or press return. If a password is requested, type in "NPS".
- After this operation is completed, quit the "Stuffit" program, and return to the finder.
- Delete the "Stuffit" program.
- Now inside the folder named "PMS_install", there will be another folder named "PMS". If Argos is already in the system, move all the files of this

folder into the Argos folder, and delete the empty folders, "PMS" and "PMS_install". If the PMS module is not going to be run through Argos and is going to be used as a stand alone application, remove the "PMS" folder from the "PMS_install" folder and delete the "PMS_install" folder.

- Substitute the "Home" stack of the system with the "Home" stack provided in this folder. If you do not want to substitute the systems "Home" stack, use a resource editor, such as "RESEDIT", to move all the resources of the "Home" stack provided with the PMS module, to the system's "Home" stack. Set the user level at the "Home " stack to "scripting".
- After all of the above steps have been followed, the person installing the system should open the PMS_driver stack, enter the password "me" and when he is prompted for a last name, he should type in the word "installer". Then he can go and set up a new card for himself as SuperUser, and delete this "installer" card.

The PMS driver

General description

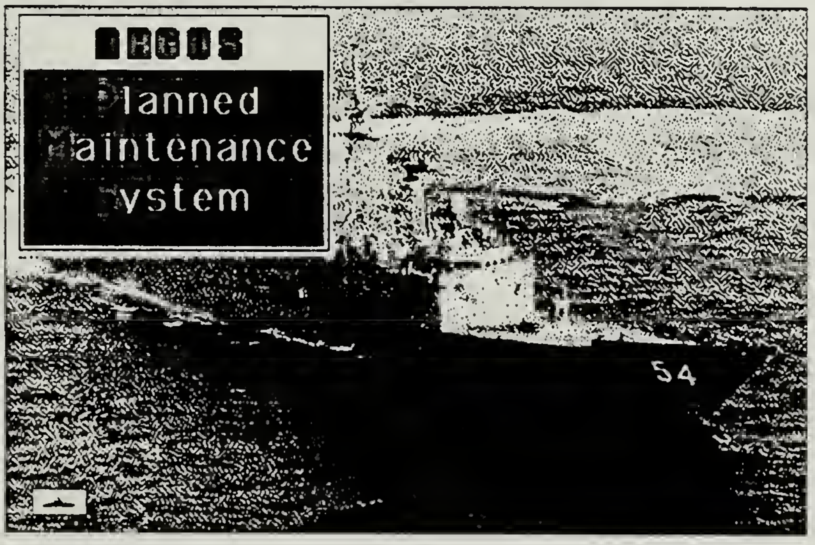
The PMS driver, is the entry point into the PMS system. It controls and restricts access to the various modules depending on the user's clearance. It could be compared to the bridge of the ship where command originate from.

One person per ship, called the "SuperUser" has access to every function of the PMS system. In general this person should be the same as the PMS coordinator of each ship.

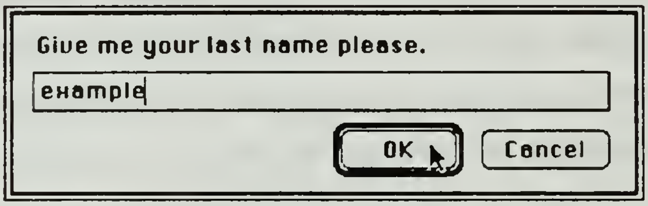
The SuperUser has the responsibility of setting up access for each individual who requests to use the PMS system. In a typical ship, every single person who belongs to the ship's crew must have access to the system, but with different access according to his position. For example, a person who belongs to the EM01 work center, should have access only to this work center. However, the engineering officer should have access to all engineering work centers. He would not, however, have access to the ship's control work centers.

Logging in the PMS system

When the PMS function of Argos is invoked, the user is presented with the following screen.

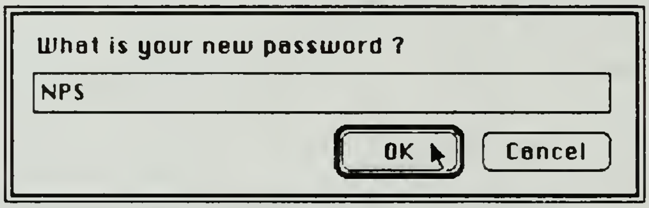


Every person logging into the PMS system for the first time, assuming he has been entered in the system by the SuperUser, is assigned the generic password "me". Upon entry of the password , "me", the system will prompt the person for their last name and the new password they want to use. This way we exclude everybody, including the SuperUser, from knowing another person's password and entering the system with that password. After this, every time they access the PMS system, they will be prompted to enter their new password. The system will set up the options available to them on the various menus of the system, and exclude them from everything else. The screens for logging into the system for the first time are shown below.


Give me your last name please.

example

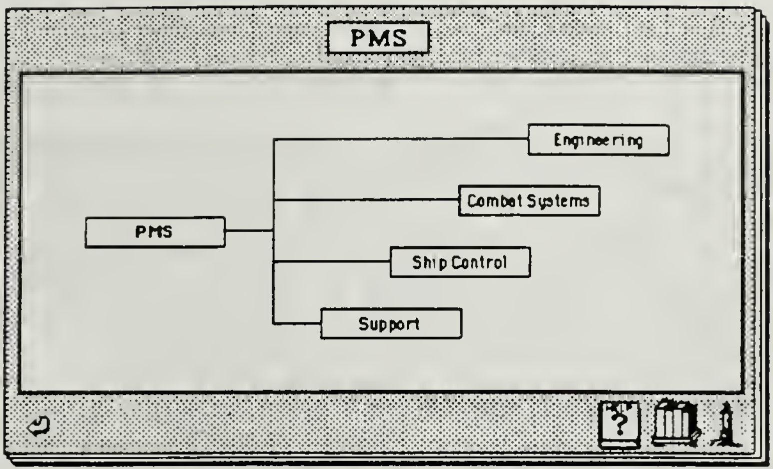
If "me" was entered as the password, the user is prompted for their last name and the new password.


What is your new password ?

NPS

After the system sets up the new password, it again prompts the user for their password, allowing them to enter the system normally.

If a person is not logging in for the first time, and have already set up a password for themselves, the above procedure is bypassed and after the password is entered, the driver's menu screen is automatically shown . At this point, they have access to the functions of the system. This screen is shown below.



In the PMS menu, the user can directly perform the functions of the PMS system. Many of these functions are also available other parts of the program, where they apply directly. For example, the print function of an MRC card is available both at the card level, and from the driver. This eliminates the need to go to the card in order to print it.

Under the division menus, the user can bring up today's daily activity card for the work center they have selected, provided of course they have access to it.

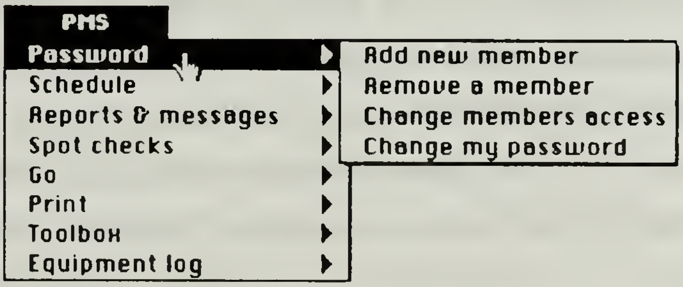
In all the menus throughout the system, functions unavailable to the user appear dimmed and cannot be selected.

The menus

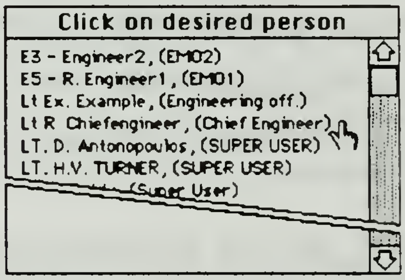
The PMS menus

The PMS menu is a two level menu system. A selection in level 1 gives us additional choices relevant to that particular selection.

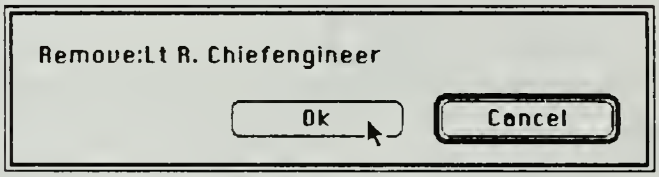
The "Password" choice: When this item is selected, we have four sub-items to choose from.



- *Add new member:* This item is available only to the SuperUser of the system. When selected, a new entry is generated in the password database. An empty card is be presented to the SuperUser, in order to allow him to fill in the required information for the new user. The generic password "me" will automatically be inserted for this person and will remain until they change it. For more information on the use of the password stack, see the appropriate section of this manual.
- *Remove a member:* This item is also only available to the SuperUser. It is used to remove a person from the system. A sorted list of all the users in the database is presented to the SuperUser so that he can select the user for removal from the system.


Click on desired person
E3 - Engineer2, (EM02)
E5 - R. Engineer1, (EM01)
Lt Ex. Example, (Engineering off.)
Lt R Chiefengineer, (Chief Engineer)
LT. D. Antonopoulos, (SUPER USER)
LT. H.V. TURNER, (SUPER USER)
(Super User)

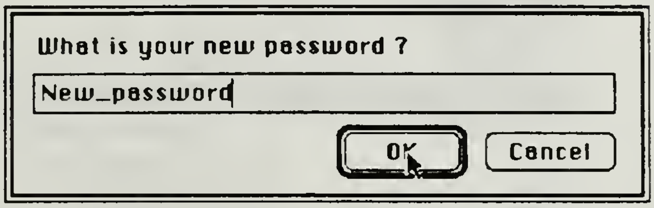
The SuperUser clicks on the name of the person he wants to remove, and the system asks to verify his choice.


Remoue:Lt R. Chiefengineer

If the SuperUser decides to proceed, the person is removed from the database, and his card is deleted from the system. If he changes his mind or has selected the wrong person, he can cancel this instruction by selecting the appropriate box.

- *Change member's access:* Item available only to the SuperUser. It is used to change a members access to the various parts of the system. This would be used in cases where a user changes positions on board the ship or he is assigned to a different division. The selection, by which person's access is changed, is done through the same procedure that is used for the *Remove a member* selection. After a person is chosen, their personal card is brought to the screen, allowing the SuperUser to make the necessary changes.

- *Change my password:* This option is available to ever user with access to the PMS database. It can be selected at any time, allowing the user to change their password. After selecting this option, the user is prompted for their password. This allows the system to verify that the person issuing the request is the same as the one who is entered in the system. If this check succeeds, the user is prompted by the following dialog box, to enter their new password twice.

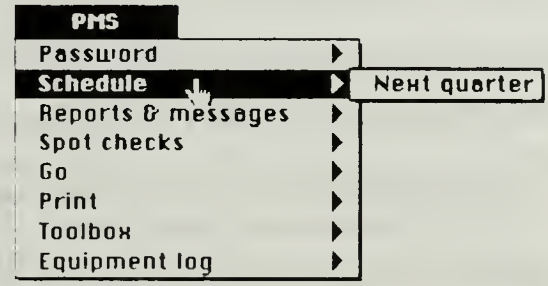


What is your new password ?

New_password

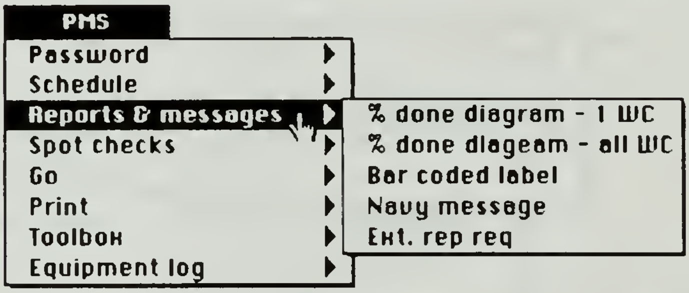
If the two passwords match, the password is changed and the user is requested to reenter the system with his new password.

The "Schedule" choice: Automatic scheduling function of the system.

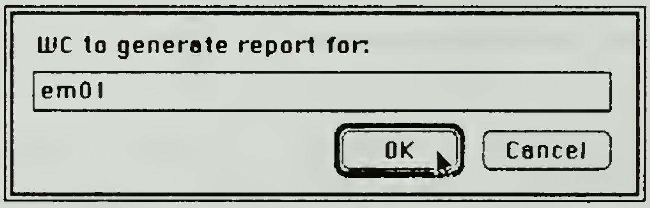


- *Next quarter:* This item is available only to the SuperUser of the system. It is provided to allow the SuperUser to start the automatic scheduling of PMS jobs for the entire ship. When selected, a calendar for the next quarter is generated and presented to the SuperUser. Using this calendar, the SuperUser can set up the ship's schedule for the quarter. This is a vital function of the system. When the scheduling is started, the system takes in account the ship's status for every day of the quarter. It uses this information to make the best possible judgement for deciding whether a particular day is suitable for the planning of PMS jobs . The calendars are stored in the "Planning" stack and details about its use are presented in the the appropriate section of this manual. Once the calendars have been generated, the user must instruct the system to begin the scheduling process.

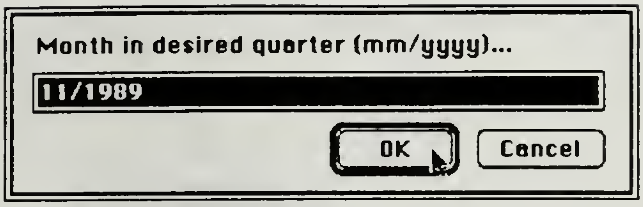
The "Reports & messages" choice: Under this menu choice the user can print feed back material on the PMS system, or other computer generated forms. In this section only basic capabilities are supplied, but other forms can be included when the need arises.



• *%done diagram - 1 WC:* This item is available to all the users involved in the supervision of the PMS system. At any time, a user can request the system to print a graph, either on screen or to the printer showing the percentage of PMS jobs completed per day , for a work center, provided that he has access to this work center. When selected the system presents him a dialog and requests the name of the work center to produce a report for. The name can be entered either in upper or lower case.

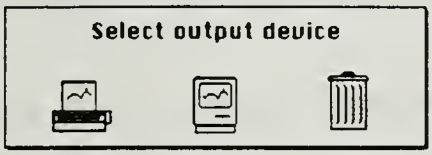

WC to generate report for:
em01

When the name of the work center has been entered, the system checks if such a work center exists and whether the user that requested the report, has access to it. If the checks succeed, the user is prompted to enter a month in the desired quarter, given as a default, the current month.


Month in desired quarter (mm/yyyy)...

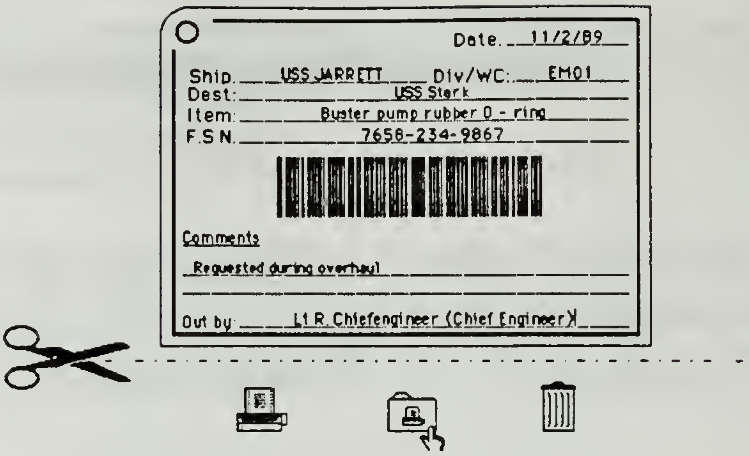
11/1989

After the month has been filled in or the default was selected, the system does a check to find out whether this quarter is in the system, and if it is requests from the user to specify the output device. If the 🗑 has been selected, the operation is canceled.



If the user selects the screen to be the output device, he is transferred to the stack that the report resides when completed, and he has the option to manually print one or all the cards individually. If the printer is selected, the output is redirected to the printer, and the user never leaves the PMS driver's card. After the operation is completed a tune is played, in order to get the user's attention if he was doing something else at that time. More information on the usage of this card is found at the corresponding section of this manual.

- *%done diagram - all WC:* This item is available to the SuperUser only. It produces exactly the same diagrams as described above but for all work centers . This function is available to supervisors only. It works exactly as the one described above, but produces a report for all available work centers.
- *Bar coded label:* This item is an utility available to every user. It produces a bar coded label to accompany an item that has to be removed from the ship for some reason. Specifically, if a piece of equipment has to go out for repair, be returned as destroyed, or is loaned to another ship, an accompanying label is produced. It is used to identify and keep track of the piece of equipment.



Date. 11/2/89

Ship. USS JARRETT Div/WC: EM01

Dest: USS Stark

Item: Buster pump rubber O - ring

F.S.N. 7658-234-9867

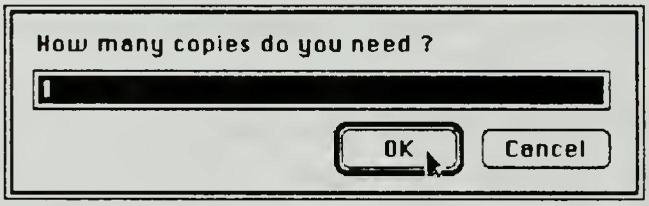
Comments

Requested during overhaul

Out by: Lt R. Chiefengineer (Chief Engineer)

When the label is presented, the user fills in the appropriate information. When the FSN is filled for the piece of equipment, the bar coded representation of the FSN is automatically generated. In case the FSN is not the information needed in bar coded form , the system can be easily modified to generate that instead. Once complete, the user has three choices. These are represented by the three buttons below the card.

The printer button is used to print the bar coded label. The user is presented with the dialog box shown below and is requested to specify the number of copies to be printed.

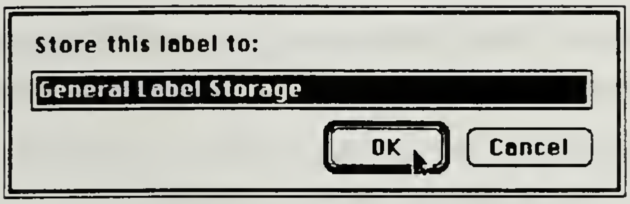


How many copies do you need ?

1

The system, by default, prints one copy.

The printer in folder button provides the capability to store the label in a stack for later reference. The user can specify the name of the file to store the label stored or it can be stored in the general file. This file is provided as a default and is called "General Label Storage".



Store this label to:

General Label Storage

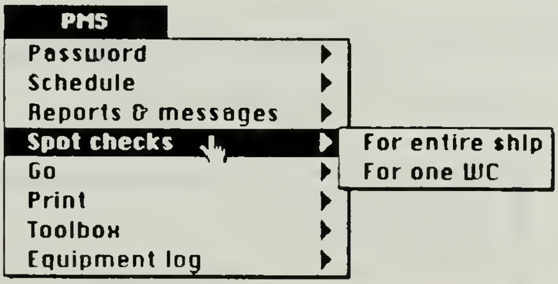
This capability is provided to allow a ship to organize the equipment going out from the ship. Specially, it can used to categorize them either by work center or repair shop to which are going to. Table representation of all the cards in a particular file can be easily produced by using Hypercards' built in "Print Report..." function selected from the "File" menu.

This button allows the user to throw away a label if desired.

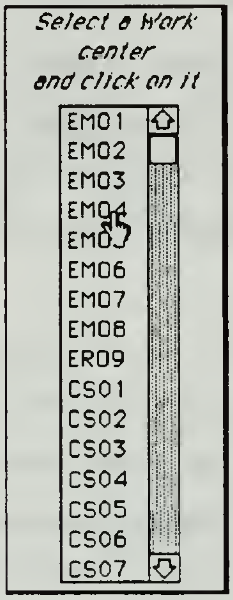
After one of the above described operations is selected, the system returns to the driver's menu card.

- *Navy message:* This item has not been implemented, because it is specific for the two Navys. It can be implemented for application to the needs for each of the Navys.
- *Ext. rep. req:* This item is available only if the system is used under Argos. When selected it returns to the Argos program allowing the user to select an item and print the computer generated request form.

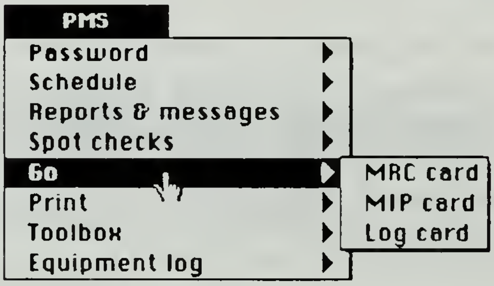
The "Spot checks" choice: This menu selection allows the system to select, at random, PMS jobs that have been completed from the quarters' daily activities cards. A spot check sheet is generated automatically for a selected work center if "For one WC" is selected or for all the ships' work centers if "For entire ship" is selected.



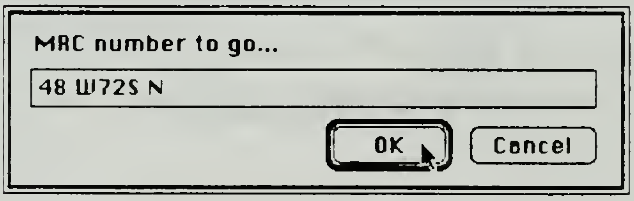
If the user selects "For entire ship", then the system will automatically start printing spot check sheets for each work center. If "For one WC" is selected, then the user will be prompted with a list of work centers from which to select one.



The "Go" choice: This menu choice allows a user to directly view a card in the PMS system, provided the number of the card is known. This selection bypasses the need of going to the corresponding database and performing a search.



- *MRC card:* When this item is selected the user is presented with the dialog box, shown below, to type in the MRC# of the desired card.

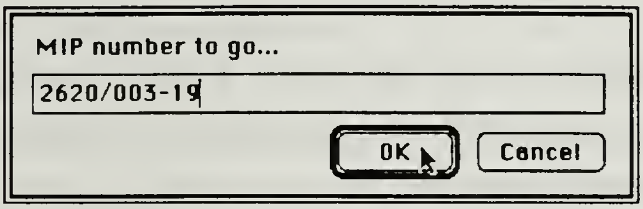


MRC number to go...

48 W72S N

After typing in the desired MRC number, the system presents the appropriate card. If the card number is wrong, or the card is not in the database, the system informs the user and returns to the driver.

- *MIP card:* When this item is selected, another dialog box prompts the user to input the desired MIP number, and the same procedure used for the MRC card is followed. The user is presented with the dialog box below.

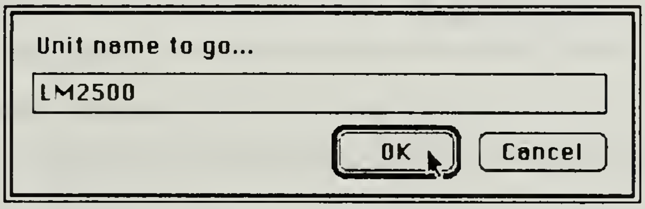


MIP number to go...

2620/003-19

Failure to find the card returns the user to the driver.

- *LOG card:* When this item is selected the user is prompted to enter some description of the Unit name. The exact name is not required for the desired unit, but only a few of the characters included in the unit name.

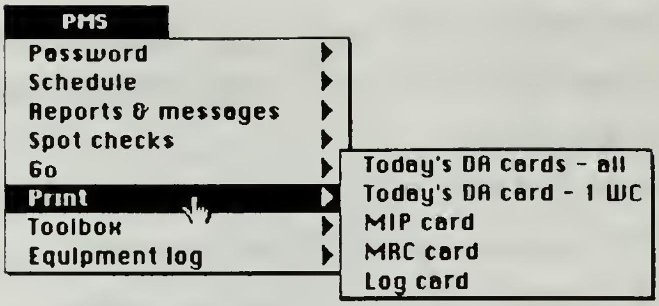


Unit name to go...

LM2500

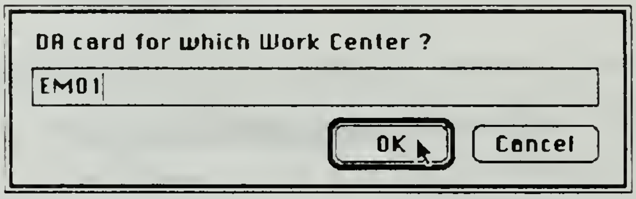
When a string like the above example is typed in, the system searches for the inclusion of the string in the "unit" field of the log database. If a match is found, the log card is presented. If the search fails, the user is informed and the request is canceled.

The "Print" choice: This choice provides to the user with a way of obtaining a hard copy of each of the items listed in the sub-menu.



- *Today's DA cards-all:* When the system performs this function, it prints out a copy of the Daily Activity cards for all the work centers for the current day. This function is used at the beginning of each day, so that the division officers can pass them out to their work center heads. A includes a description of the jobs that the work center has to completer that day. Prior to this function being used, automatic rescheduling of uncompleted jobs takes place. This allows the system to move forward all jobs that were not completed the previous day. This function is available only to the SuperUser.

- *Today's DA cards-1 WC:* This function is available to every user. It provides a printout of the the Daily Activity card for a selected work center. It is used both as a reference for the division officer and as a report for what jobs have been completed on a particular day. The user is presented with the following dialog box. The user types in the name of the work center for which a printout of the Daily activity card is needed .


DA card for which Work Center ?

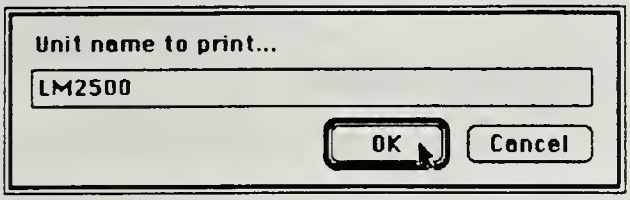
EM01

After typing in the name of the desired work center, the system checks whether or not the user has access to this work center and proceeds only if they do. If access is not available or the user changes their mind, the system informs them and exits the request.

- *MIP card:* When "MIP card" is selected to be printed, the system follows the same procedure followed for the "Go" menu item described above.

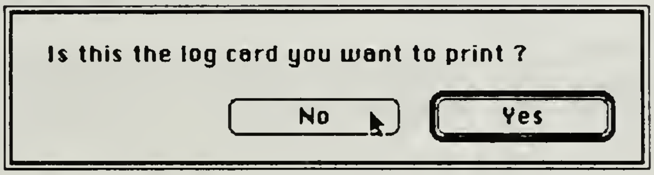
- *MRC card:* When "MRC card" is selected to be printed, the system follows the same procedure followed for the "Go" menu item described above.

- *LOG card:* When "LOG card" is selected, the system shows a dialog box allowing the user to input a search string for the desired unit. This dialog is presented below.

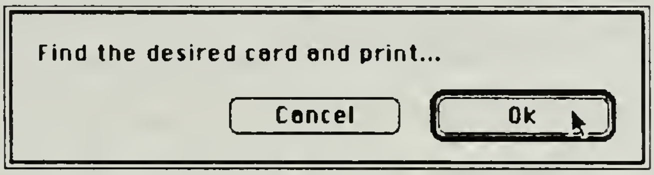

Unit name to print...

LM2500

When the user types in the desired unit or part of it, the system does a search in the Log database. In the case of no match, feed back information is presented and the request is exited. In case a match is found and the card is displayed on the screen, a new dialog box is presented requesting the user to ensure that this is the correct card. This is done because machinery does not have unique names, unlike PMS cards , and multiple log cards for the same kind of equipment can exist.

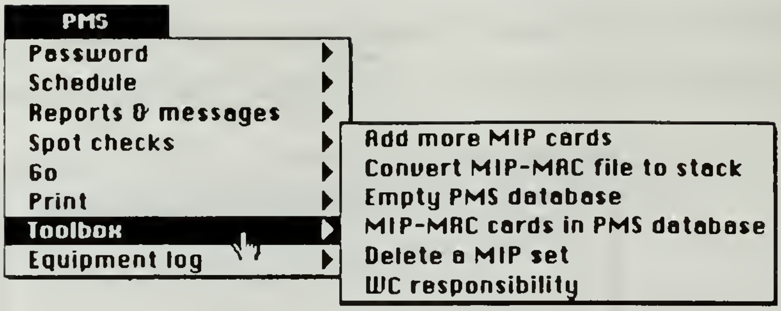

Is this the log card you want to print ?

In case this is not the required card, the user is prompted to find the desired log card and do a manual print by pressing the print button of the card found.


Find the desired card and print...

From here the user can do a search on a different field, or with a different search key.

The "Toolbox" choice: This choice is provided only to the SuperUser of the system. It gives him the ability to perform global operations on the PMS database.



- *Add more MIP cards:* When this item is selected, the SuperUser is presented with the corresponding card of the "Toolbox" stack. Using this tool new cards can be added to the PMS database, provided they are in the correct format. The new cards must be included in a stack. Tools to transform the cards to the correct format are provided below in the same menu. Specific instructions for this operation are given in the "Toolbox" section of this manual.

- *Convert MIP-MRC file to stack:* This item gives to the SuperUser the capability to convert a MIP-MRC text file read in from the on line PMS database to a MIP-MRC stack. The files coming in by phone, should be text files without control characters. This instruction prepares the updated versions of PMS cards to be entered into the system, eliminating the typing process. It must be run first so as to allow the SuperUser to use the function described above.. Specific instructions for this operation are given also in the "Toolbox" section of this manual.

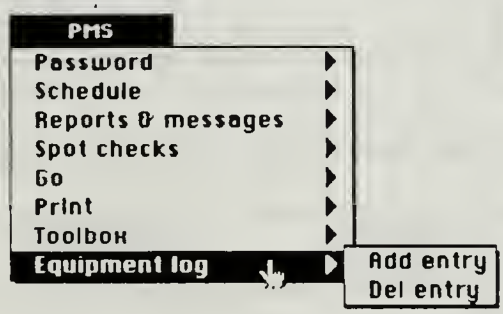
- *Empty PMS database:* This item gives the user the capability to empty the entire PMS database, allowing generation of a new one in case the system is to be used for another class of ship. It is available only to the SuperUser and uses a series of check points to ensure that the person who has instructed the system to do so, is the SuperUser. Specific instructions for this operation are given also in the "Toolbox" section of this manual.

- *MIP-MRC cards in PMS database:* This item produces a list of all the MIP and MRC cards in the PMS database for reference purposes. Specific instructions for this operation are given also in the "Toolbox" section of this manual.

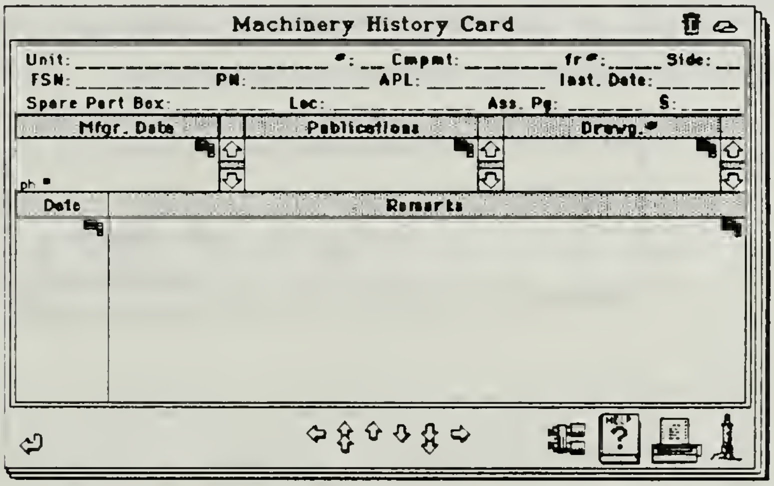
- *Delete a MIP set:* This item provides the SuperUser with the capability to delete a MIP card and all its corresponding MRC cards from the PMS database. It is used to replace an existing when an updated one comes in. Specific instructions for this operation are given also in the "Toolbox" section of this manual.

- *WC responsibility:* This function assigns the relevant work centers to the corresponding MRC cards. It is used when new cards are added to the system or when a work center is added or removed from a particular MRC job. Specific instructions for this operation are given also in the "Toolbox" section of this manual.

The "Equipment log" choice: This choice is available to all division officers. It gives them the capability to add a new entry to the log database or remove an existing entry in cases where a piece of equipment is removed from the ship.

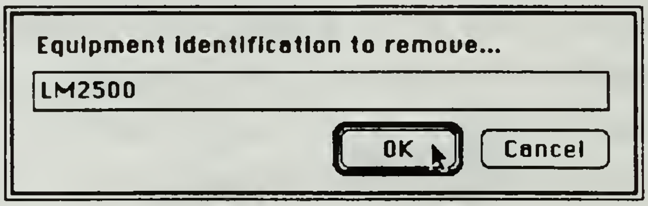


- *Add entry:* When this item is selected, the user is presented with a fresh copy of a log card, to fill out.



The only difference from the cards presented in the log stack, is the the capability to delete the card by pressing the 🗑 icon. This icon is not available in the log stack except in case of a manual deletion as described below. After the user fills in the desired information, he can either store the card in the log database by pressing the ☁ button or throw it away with the 🗑 button.

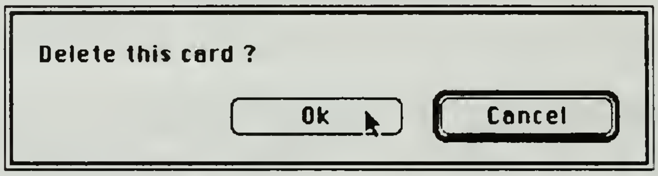
- *Del entry:* By selecting this option the user has the capability to delete a card from the log stack. The user is prompted, with the dialog box that appears below, to enter some identification for the equipment that has to be removed from the system. By doing so he invokes a search command. If the search fails and no match is found, feedback information appears on the screen, and the request is ended.



Equipment Identification to remove...

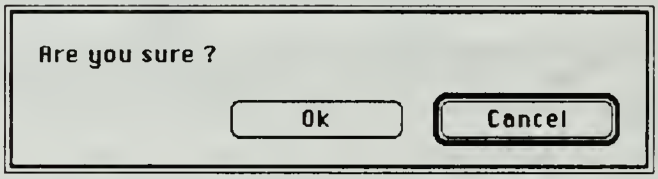
LM2500

If a match is found the system presents the card where the match was found, and asks to verify that the card found is the one to be deleted. This dialog box follows.



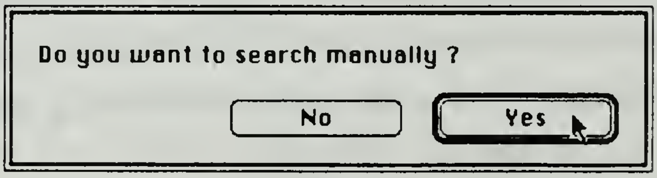
Delete this card ?

If "Ok" is selected, the system asks the user to once again verify for his selection. This is done because the deletion of a card cannot be undone.



Are you sure ?

If the user replies that this is not the card to be deleted, he is prompted to do a manual search, either by finding the card, or invoking a local search according to some other search pattern. At the same time, the 🗑 button appears on the screen, enabling the manual delete function. The corresponding dialog box follows.

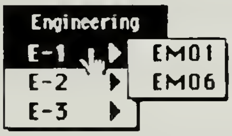


Do you want to search manually ?

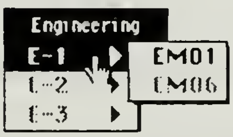
When the desired card is found, the user is returned to the driver's menu card, to proceed with some other action.

The division menus

There are four division menus. Each is divided into sub-menus for the various work centers. From these menus, the user goes directly to the current daily activity card for their work center. According to the access and position on board the ship a user has, the corresponding menus are activated. Samples of those menus are given below.



Engineering officer's menu



EM01 personell menu

As shown in the diagram above, the Engineering officer has access to all the engineering work centers. On the other hand, a member of one work center has access only to their work center. Menu items that appear dim cannot be selected.

The PMS Toolbox

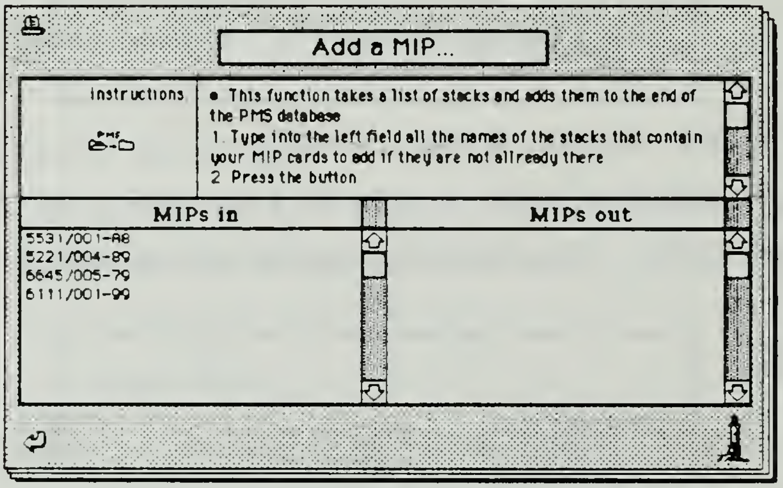
General description

The PMS Toolbox is a collection of programs that lets the SuperUser perform global manipulation of the PMS stack. Each card of this stack performs a different function. Instructions for using each function are provided on the card as help to the user.

The "Toolbox" stack, at present, consists of six tools. As the need arises, additional tools can be easily added.

The PMS Toolbox Tools

The "Add a MIP..." card: This function provides the user with the capability to add new PMS cards to the PMS database.



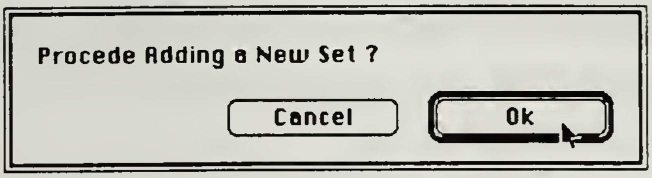
- This function takes a list of stacks and adds them to the end of the PMS database

1. Type into the left field all the names of the stacks that contain your MIP cards to add if they are not allready there
2 Press the button

5531/001-A8
5221/004-89
6645/005-79
6111/001-99

The user enters the names of the stacks to be added to the database in the "MIPs in"field . These stacks can consist of multiple MIP and multiple MRC cards.

PMS ⇒ This button starts the execution of the "Add a MIP..." function . A message informing the user that the program is executing appears at the bottom of the screen . The cursor changes from ☝ to ⌚. When this is done, a dialog box appears, asking the user to verify his choice. This dialog box is shown below.

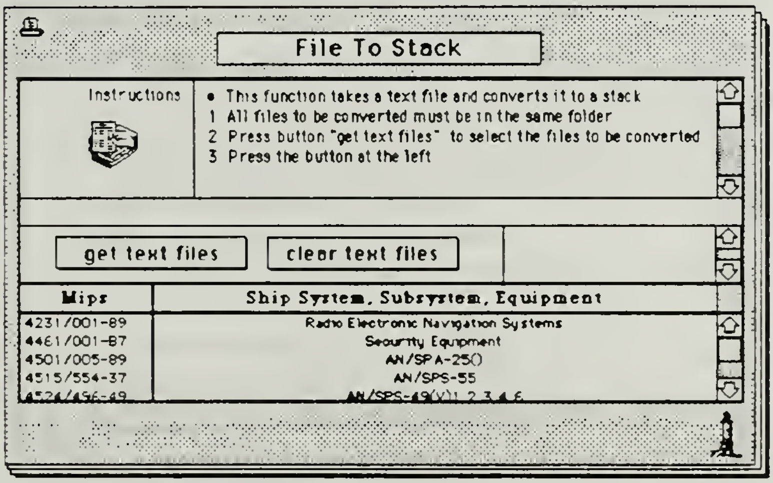

Procede Adding a New Set ?

As the stacks are processed, they are removed from the left field and added to the "MIPs out" field. When the operation is complete, the "Up to Date" field of the "MIP/MRC List" card is set to "NO" to inform the user the PMS database list is updated.

This button returns the user back to the PMS driver.

This button lets the user to exit the PMS system and return to the login card of the driver.

The "Convert MIP-MRC file to stack" card: This choice provides the user with the capability of converting PMS text files, consisting of MIPs and MRCs, into PMS MIP and MRC cards.


File To Stack

Instructions

- This function takes a text file and converts it to a stack

1 All files to be converted must be in the same folder
2 Press button "get text files" to select the files to be converted
3 Press the button at the left

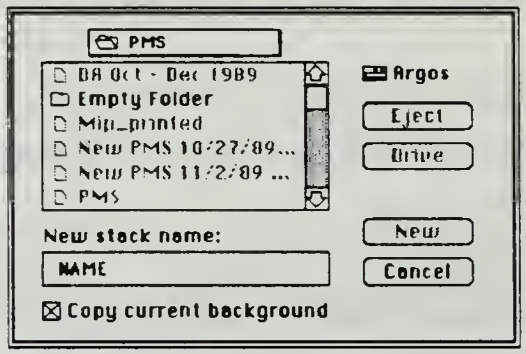
Mips	Ship System, Subsystem, Equipment
4231/001-89	Radio Electronic Navigation Systems
4461/001-87	Security Equipment
4501/005-89	AN/SPA-25()
4515/554-37	AN/SPS-55
4524/456-49	AN/SPS-49(V)1,2,3,4,6

get text files

To start the process of converting text files to cards, the user must use the "get text files" button to select those files that they wish to convert. Upon selection of the button, the user is shown a standard

Macintosh file selection dialog box from which to select those files to be converted. The files selected will be shown in the field to the right of the "clear text files" button.

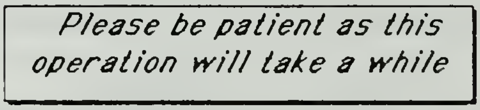
Once all text files have been selected ,then the user will press the button at the left to initiate the transformation process. Once the process is started, then the user will be presented with the following new stack dialog box.



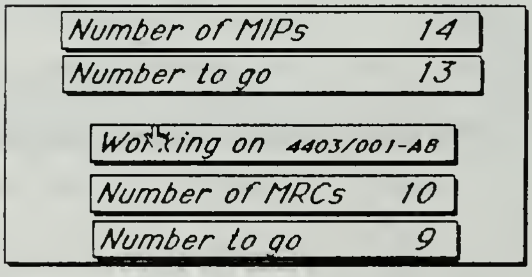
New stack name: NAME

☒ Copy current background

At this point the user should enter any letter or number into the dialog box and then press New. Once this is complete, the system will start reading the first text file and then will show the following information fields.



Please be patient as this operation will take a while



Number of MIPs 14

Number to go 13

Working on 4403/001-AB

Number of MRCs 10

Number to go 9

As the individual text files are read, the appropriate MIP number in shown in the "working on" field. Additionally the MIP number is entered into the list of MIPs.

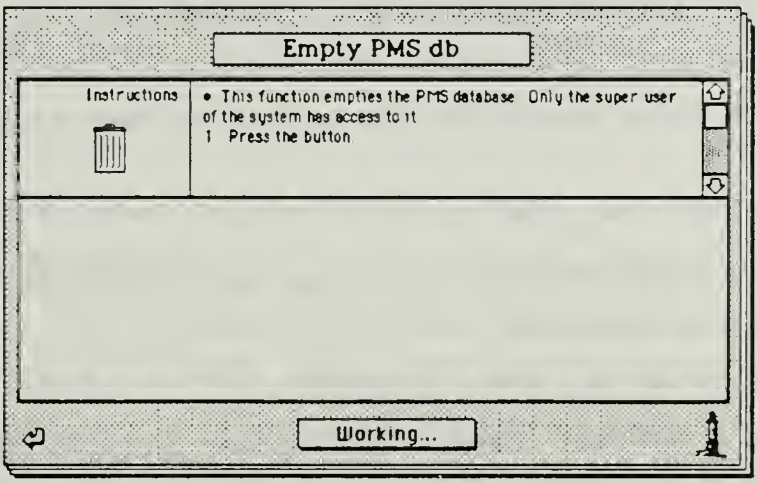
clear text files

Once the operation is complete, the user can use the "clear text file" button to clear the list of text files that have been converted to cards.

This button returns the user back to the PMS driver.

This button lets the user to exit the PMS system and return to the login card of the driver.

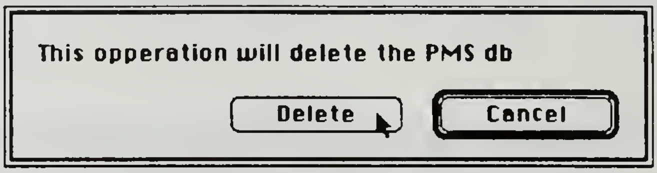
The "Empty PMS database" card: This function provides the SuperUser with the capability to empty the entire PMS database including all MIP and MRC cards. Six empty cards, that form the templates for the addition of new cards later on, are the only cards that are left in the database. This function is extremely dangerous. Additional check points are provided to ensure that the person requesting the execution of this function is the SuperUser. The card for this function is shown below.



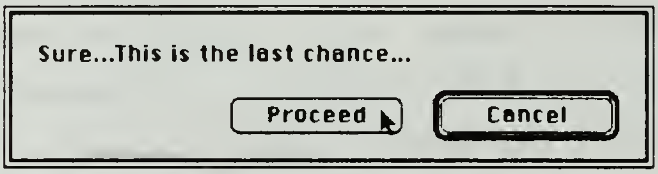
• This function empties the PMS database. Only the super user of the system has access to it

1 Press the button

This button initiates the of execution of the "Empty PMS db" function . When pressed, the user is prompted to enter their password. The system checks whether or not this person has been assigned the position of SuperUser in the Password stack and whether or not the password entered is the same as the one he used to log into the system. This provides a margin of safety by the system to prevent errors. If the check succeeds, the user is prompted with the following dialog box to verify his choice.


This opperation will delete the PMS db

And again...

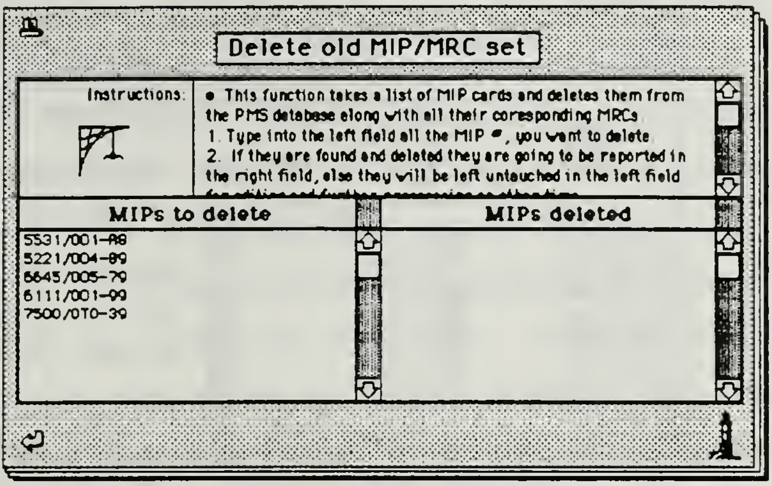

Sure...This is the last chance...

If the checks and warnings are successfully passed, a message informing the user that the program is executing appears at the bottom of the screen and the cursor changes from ☝ to ⌚.

This button returns the user back to the PMS driver.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "Delete a MIP set" card: This function provides the SuperUser with the tools to remove a MIP card along with its corresponding MRC cards. In case a MIP card is to be replaced by a newer version, this function is used to delete it. This function is set up this way so that even if a new version of the card is added to the system before the deletion of the old one, only the old one is deleted.



• This function takes a list of MIP cards and deletes them from the PMS database along with all their coresponding MRCs
1. Type into the left field all the MIP #, you want to delete
2. If they are found and deleted they are going to be reported in the right field, else they will be left untauched in the left field

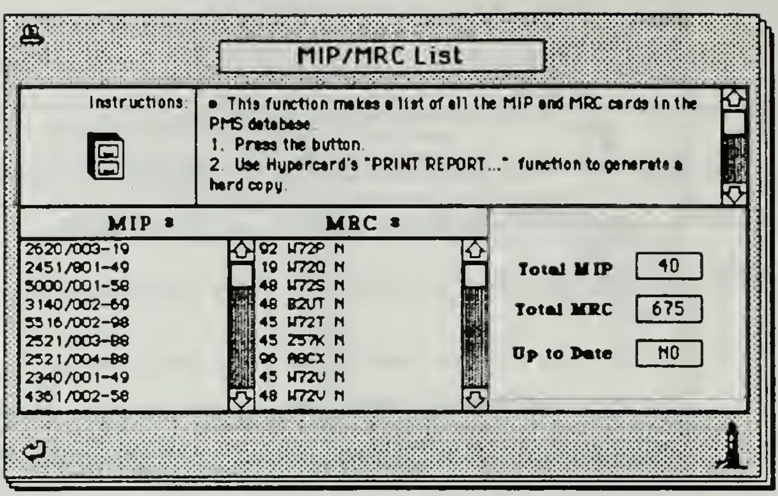
MIPs to delete:
5531/001-A8
5221/004-89
6645/005-79
6111/001-99
7500/0T0-39

This button initiates execution of the "Delete a MIP set" function . A message informing the user that the program is executing appears at the bottom of the screen and the cursor changes from ☝ to ⌚. As the execution proceeds, to give some feed back to the user, the MIP names are deleted from the "MIPs to delete" field and added to the "MIPs deleted" field. In the case where a MIP card is not found, it remains in the "MIPs to delete"field.

This button returns the user back to the PMS driver.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "MIP-MRC cards in PMS database" card: By using this tool, the user has the capability to produce a list of all the MIP and MRC numbers in the PMS database. The time of execution of this operation varies from a few minutes to a couple of hours, depending of the size of the PMS database. To produce a hard copy of the results, the user can use the "Print Report" command of HyperCard.



Instructions:

- This function makes a list of all the MIP and MRC cards in the PMS database.

1. Press the button.
2. Use Hypercard's "PRINT REPORT..." function to generate a hard copy.

MIP #	MRC #
2620/003-19	92 W72P N
2451/801-49	19 W72Q N
5000/001-58	48 W72S N
3140/002-69	48 B2UT N
5516/002-98	45 W72T N
2521/003-88	45 Z57K N
2521/004-88	96 ABCX N
2340/001-49	45 W72U N
4361/002-58	48 W72V N

Total MIP: 40

Total MRC: 675

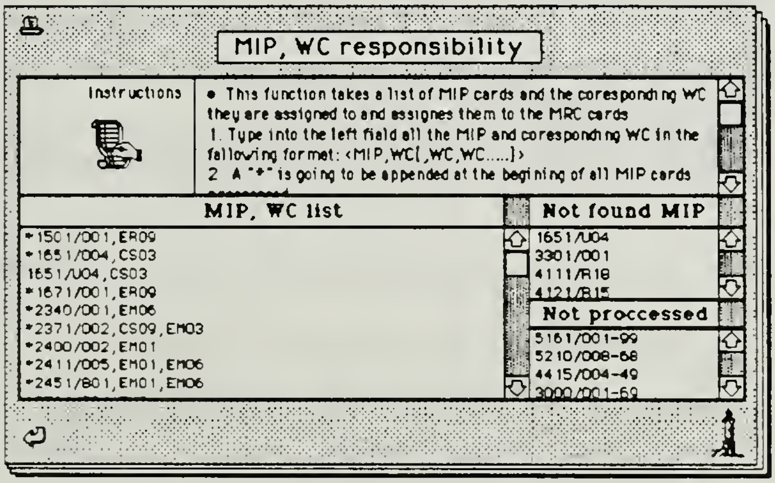
Up to Date: NO

This button initiates the execution of the "MIP/MRC List" function . A message informing the user that the program is executing appears at the bottom of the screen and the cursor changes from 👆 to ⌚. The MIP cards found are entered in the "MIP #" field and the MRCs in the "MRC #" field . The total number of MIP and MRC cards is shown in the corresponding fields. The "Up to Date" field is set to "YES".

This button returns the user back to the PMS driver.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "WC responsibility" card: This function is used after a new MIP set has been inserted into the PMS database. It assigns the work centers that are responsible for the MRC cards. This is done so that when the system does the automatic scheduling, it takes this information into account and schedules the same job to be done by all the work centers involved. For example, if a MRC job requires two engineers and one electrician to perform it, the system schedules both the corresponding engineering work center and the electrical work center. All MIP cards of a ship must be inserted into the "MIP,WC, responsibility"field. The format used to enter them is <MIP #,WC1,WC2,WC3......>. If responsibility for some MIP cards is modified, for example to add a new work center or to delete one for a particular piece of equipment, the SuperUser deletes the "*" from the beginning of the line the MIP is on and all the MRC's for this MIP are reprocessed. When a MIP is processed successfully, an asterisk is appended at its beginning. This allows control on which MIPs are going to be processed. Also, execution becomes much faster, because only the required or previously processed MIP cards are examined and updated. In case a MIP is not in the system, it is reported in the "Not found MIP" field . After the assignment of jobs is done, the system searches through the entire PMS database, and pulls out all the MIP cards for which no work center is assigned responsibility . This provides feed back to the SuperUser, showing duplicated MIPs in the PMS database, because work centers are only assigned once for each MIP number in the system. This avoids multiple scheduling of the same job.



- This function takes a list of MIP cards and the corespondi ng WC they are assigned to and assignes them to the MRC cards
1. Type into the left fiald all the MIP and coresponding WC in the fallowing formet: <MIP,WC[,WC,WC.....]>
2 A "*" is going to be appended at the begining of all MIP cards

*1501/001,ER09
*1651/004,CS03
1651/U04,CS03
*1671/001,ER09
*2340/001,EM06
*2371/002,CS09,EM03
*2400/002,EM01
*2411/005,EM01,EM06
*2451/B01,EM01,EM06

1651/U04
3301/001
4111/R18
4121/R15

5161/001-99
5210/008-68
4415/004-49
3000/001-69

This button initiates execution of the "MIP, WC responsibility" function . A message informing the user that the program is executing appears at the bottom of the screen and the cursor changes from ☝ to ⌚.

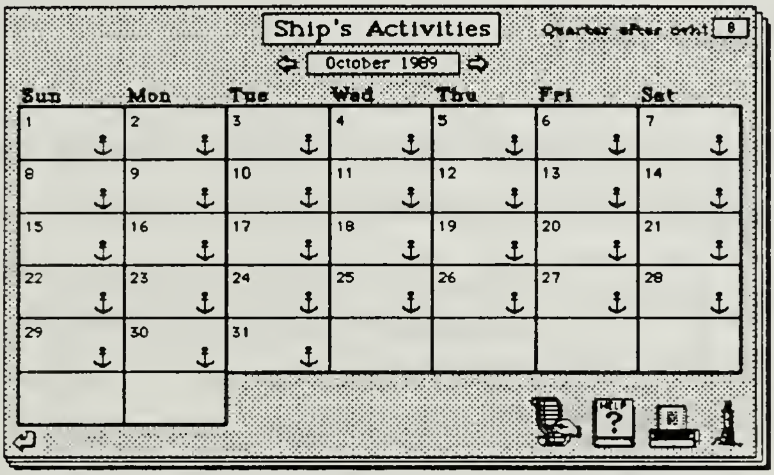
↩ This button returns the user back to the PMS driver.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "Planning" stack

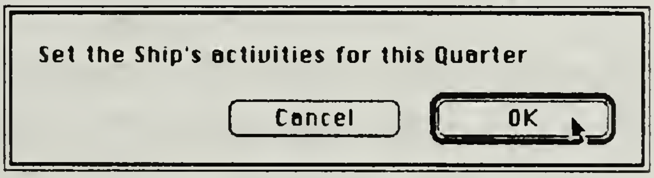
General description

The "Planning" stack is visible and available only to the SuperUser. This stack is where new calendars, for scheduling purposes, are generated and stored. When the SuperUser asks the system to schedule the next quarter, the system generates the calendars for the quarter. The user is then presented with the calendars to set the "ship's status" for every day of that quarter. A typical card for a month is shown below.



Sun	Mon	Tue	Wed	Thu	Fri	Sat
1 ⚓	2 ⚓	3 ⚓	4 ⚓	5 ⚓	6 ⚓	7 ⚓
8 ⚓	9 ⚓	10 ⚓	11 ⚓	12 ⚓	13 ⚓	14 ⚓
15 ⚓	16 ⚓	17 ⚓	18 ⚓	19 ⚓	20 ⚓	21 ⚓
22 ⚓	23 ⚓	24 ⚓	25 ⚓	26 ⚓	27 ⚓	28 ⚓
29 ⚓	30 ⚓	31 ⚓				

The user is prompted to define the ship's activities for the quarter to be scheduled.



Set the Ship's activities for this Quarter

By "ship's status" or "ship's activities", we mean the classification for each individual day. The ship's status will be taken into account when the automatic scheduling is done so as to avoid unrealistic job scheduling and job distribution. When all days are classified, the SuperUser has the capability to initiate the actual job scheduling. The days are classified by clicking on the small icon at the right hand corner of each day. The meaning of each icons is as follows:

⚓ The ship is at port, at normal duty.

The ship is underway, and only PMS jobs that can be done in such a condition should be scheduled this day.

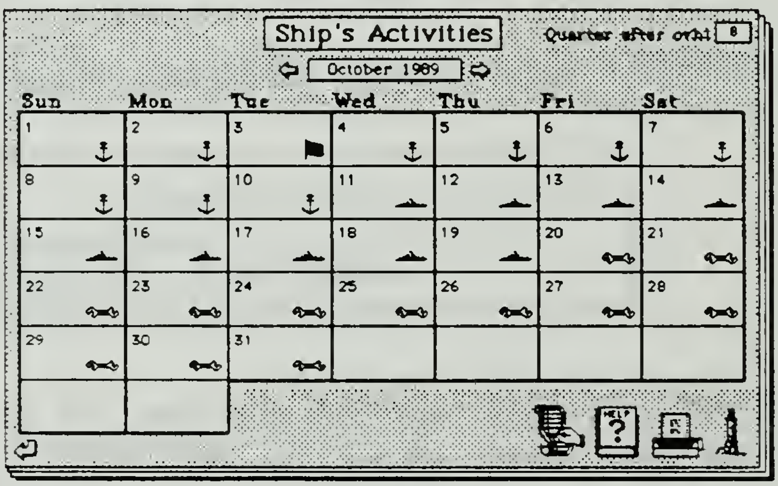
Major Holiday. No big PMS jobs should be scheduled this day, due to the unavailability of personnel.

Overhaul. Any PMS job can be scheduled during this period.

Inactive status.

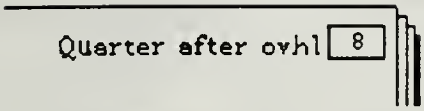
The program could be implemented to run faster, but the trade off would be worst job distribution and less thorough checking. The reason the decision was made to implement it this way was, that as hardware improves and faster machines are introduced, the performance of the system will directly positively be effected.

A typical calendar, showing the various days status, follows along with a detailed explanation of the available options and functions.

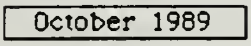


The fields

- *The Quarter after ovhl field:* In the right top corner, a field indicates to the user which quarter after overhaul is the current one.

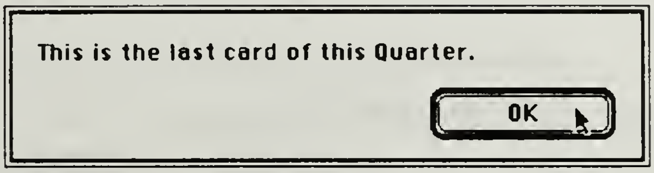


- *The Current month field:* In the middle of the screen a field indicates to the user the current month he is working on, and who's calendar is shown.



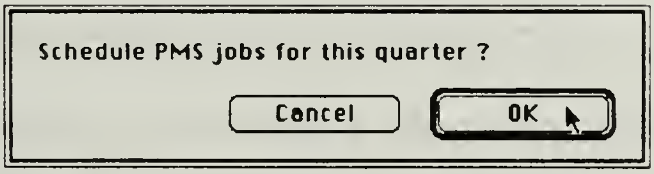
The buttons

⇨ Presents the next calendar month. In case this is the last month of the quarter, a dialog box appears, informing the user accordingly.

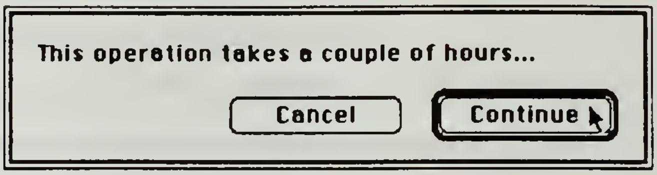

This is the last card of this Quarter.

⇦ Shows the previous month.

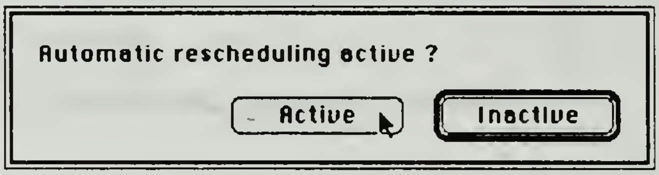
This button initiates the execution of the "Automatic Scheduling" function . When pressed, a dialog box asking the user to verify his selection appears.


Schedule PMS jobs for this quarter ?

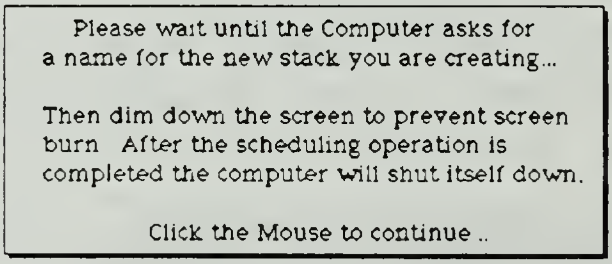
If the user changes their mind, they have the opportunity, to cancel the scheduling job. The system will continue and delete all the calendar cards for this quarter and return to the PMS driver. If the user decides to continue, they will be prompted with another dialog, to reverify their choice.


This operation takes a couple of hours...

The actual scheduling takes from a couple of hours to about two days to complete. This time depends on the system's configuration, the size of the PMS database of the particular ship, and the number of work centers involved. For an FFG - 7 class ship, it takes about two days to complete. When permission to proceed is granted, the user is presented with another dialog asking him to specify whether the automatic transfer option for jobs that were not completed one day to the next should be active.


Automatic rescheduling active ?

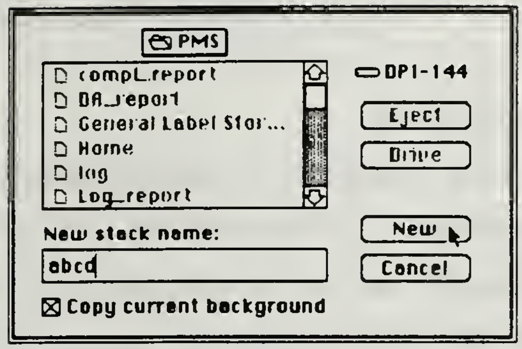
And finally a short statement of instructions is given to the user.


Please wait until the Computer asks for
a name for the new stack you are creating...

Then dim down the screen to prevent screen
burn After the scheduling operation is
completed the computer will shut itself down.

Click the Mouse to continue ..

When the mouse button is clicked, the user is presented with the standard HyperCard "New Stack..." dialog box to name the new stack to be generated.



New stack name: abcd

☒ Copy current background

In this dialog box a string of at least one character should be entered. The string can be any random sequence of characters, because the correct name of the stack is going to be set by the system. Then either "New" should be selected or the return key pressed. The system will start the automatic scheduling process. Upon completion, the computer will be shut down automatically.

The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

The print button prints the current quarter's calendar.

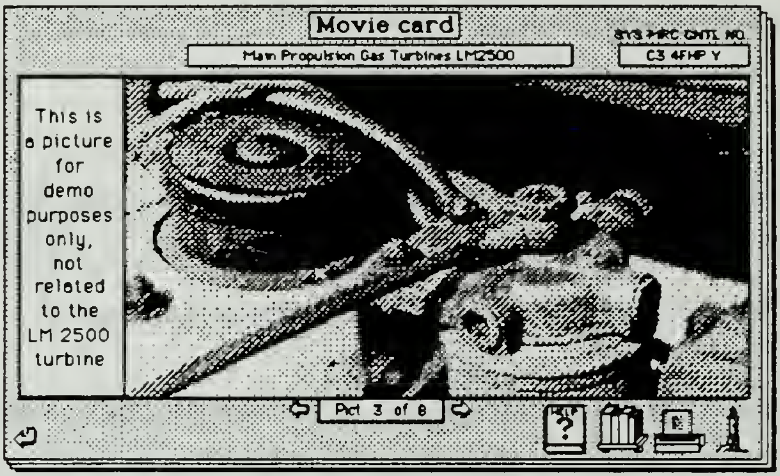
This button returns the user back to the PMS driver.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "Movies" stack

General description

The "Movies" stack is a collection of images taken with an ordinary 35mm camera and scanned into the system. Its purpose is to give the user a better feel of the maintenance to be completed, and to point out intricate parts of the actual PMS job. The images show parts that are hard to describe and/or jobs that are difficult to complete without on the spot supervision. This function will play a big role in the training of the personnel and should help them perform the maintenance jobs, unattended. It will also give them the capability to review a particular job at their own pace, plus it will give an instructor the tools to explain a job better before it is performed. A sample picture for the C3 4FHP Y MRC card is shown below.



Main Propulsion Gas Turbines LM2500

C3 4FHP Y

This is a picture for demo purposes only, not related to the LM 2500 turbine

The fields

• *The Equipment field:* At the top of each card, a field shows the user the name of the equipment to which the image belongs. This name is the same name that is shown on the top of the corresponding MRC card.

Main Propulsion Gas Turbines LM2500

• *The Sys MRC cntl no field:* On the right top corner of each picture card, a field labeled "SYS MRC CNTL NO" shows the MRC card number to which this image belongs.

SYS MRC CNTL NO

C3 4FHP Y

- *The "Pict of" field:* At the bottom of each picture card, a field informs the user how many images belong to this particular MRC job and on which image they are currently on.

Pict 3 of 8

The buttons

⇨ This button shows the user the next image that belongs to this same MRC job. On the last card for each job, this arrow does not appear.

⇦ This button shows the user the previous image. In case this is the first image of a set, the arrow is not visible.

HELP ? The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

The print button prints a copy of the picture.

↩ This button returns the user back to the MRC card they were on when they invoked the "movies" function.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "Password" stack

General description

The "Password" stack is the stack in which one card for each person that has access to the system is stored. Each user has a card, tied to his password, that contains all the information on what they have access to. This way the system restricts the user from accessing functions they are not eligible for. If somebody tries to open this stack and review or change its contents, the program forces them out the PMS system. Private access to the system thus is accomplished. The password system's card is divided to five different parts. The parts for a typical division officer, along with the description of each, are shown below. By selecting one of the five different stack buttons at the top, we gain access to the various sections of the password system. The SuperUSer then selects the items by marking the appropriate boxes that the person whose card he is working on to have access to. Each entry corresponds to an item in the various menus of the PMS system. After the desired items have been set, the SuperUser can store, print or delete the current card by pressing the respective buttons at the top of the card. The very first time the PMS system is used, a card, named "installer", is provided to allow full access to the system by the person who is installing the system . This password should be changed immediately.

The fields

• *The Rank/Name field:* contains the rank and name of the person to which the card belongs.

Rank / Name :..

• *The Position field:* contains the current position of the individual on board the ship. One card in the stack must contain the name "Super User", for the SuperUser of the system to allow him access to the functions available only to him.

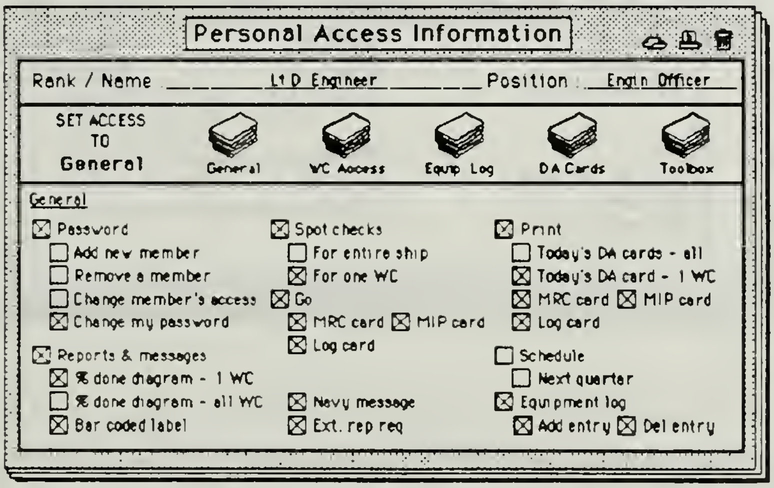
Position :..

- *The Password field:* This field is not accessible directly by anyone. It contains the password of the person to which the card belongs. The field may not be seen by even the SuperUser or the owner. The only way it can be changed is through the PMS driver menu selection. This field, initially set to "me" for every new user, can be changed the first time they log into the system.
- *The "Set Access To" field:* This field shows to the user the general category he is setting access to. The word in bold typeset, is one of the five names of the various categories.

SET ACCESS
TO
WC Access

The buttons

General This button presents to the SuperUser the general access menu. It controls access to the PMS menu of the driver.



Rank / Name Lt D Engineer Position : Engin Officer

SET ACCESS
TO
General

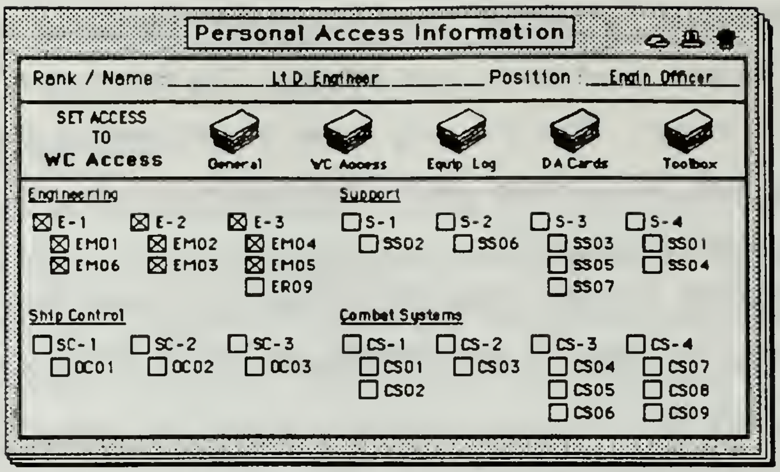
General

☒ Password
  ☐ Add new member
  ☐ Remove a member
  ☐ Change member's access
  ☒ Change my password
☒ Reports & messages
  ☒ % done diagram - 1 WC
  ☐ % done diagram - all WC
  ☒ Bar coded label

☒ Spot checks
  ☐ For entire ship
  ☒ For one WC
☒ Go
  ☒ MRC card ☒ MIP card
  ☒ Log card
  ☒ Navy message
  ☒ Ext. rep req

☒ Print
  ☐ Today's DA cards - all
  ☒ Today's DA card - 1 WC
  ☒ MRC card ☒ MIP card
  ☒ Log card
☐ Schedule
  ☐ Next quarter
☒ Equipment log
  ☒ Add entry ☒ Del entry

WC Access This button presents to the SuperUser, the work center menus. It is used to control access to the DA cards for the various work centers.



Rank / Name: Lt D. Engineer
Position: Engin. Officer

WC Access

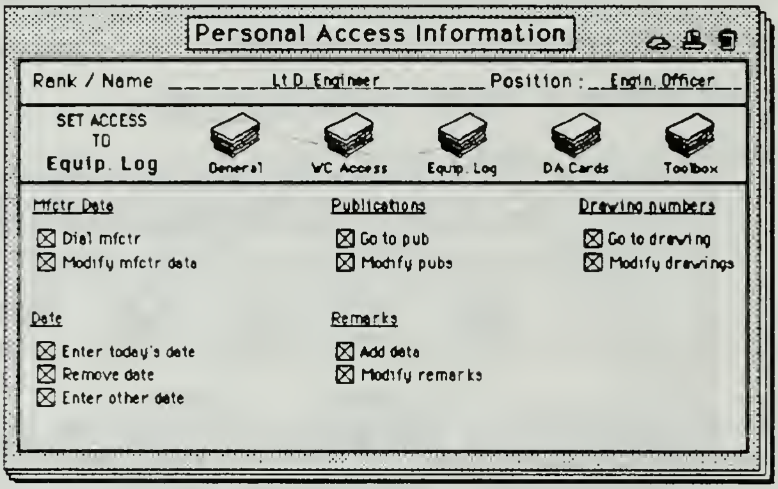
Engineering
- ☒ E-1
  - ☒ EM01
  - ☒ EM06
- ☒ E-2
  - ☒ EM02
  - ☒ EM03
- ☒ E-3
  - ☒ EM04
  - ☒ EM05
  - ☐ ER09

Support
- ☐ S-1
  - ☐ SS02
- ☐ S-2
  - ☐ SS06
- ☐ S-3
  - ☐ SS03
  - ☐ SS05
  - ☐ SS07
- ☐ S-4
  - ☐ SS01
  - ☐ SS04

Ship Control
- ☐ SC-1
  - ☐ OC01
- ☐ SC-2
  - ☐ OC02
- ☐ SC-3
  - ☐ OC03

Combat Systems
- ☐ CS-1
  - ☐ CS01
  - ☐ CS02
- ☐ CS-2
  - ☐ CS03
- ☐ CS-3
  - ☐ CS04
  - ☐ CS05
  - ☐ CS06
- ☐ CS-4
  - ☐ CS07
  - ☐ CS08
  - ☐ CS09

Equip. Log This button presents to the SuperUser the options available in the Equipment log stack. In general, access to these functions is given only to division officers and higher.



Rank / Name: Lt D. Engineer
Position: Engin. Officer

Equip. Log

Mfctr Data
- ☒ Dial mfctr
- ☒ Modify mfctr data

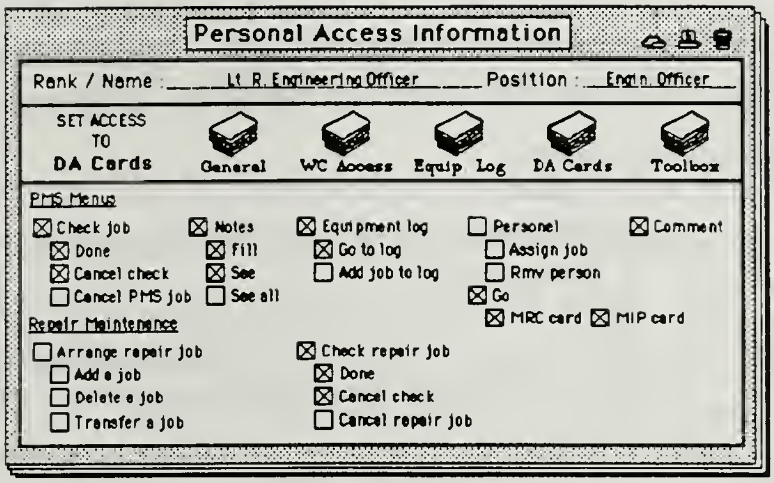
Publications
- ☒ Go to pub
- ☒ Modify pubs

Drawing numbers
- ☒ Go to drawing
- ☒ Modify drawings

Date
- ☒ Enter today's date
- ☒ Remove date
- ☒ Enter other date

Remarks
- ☒ Add data
- ☒ Modify remarks

DA Cards This button presents to the SuperUser the options for the Daily Activity cards. Every one on board ship is given access to these cards. According to the different options selected, different menus become available to them.



Rank / Name : Lt R. Engineering Officer
Position : Engin. Officer

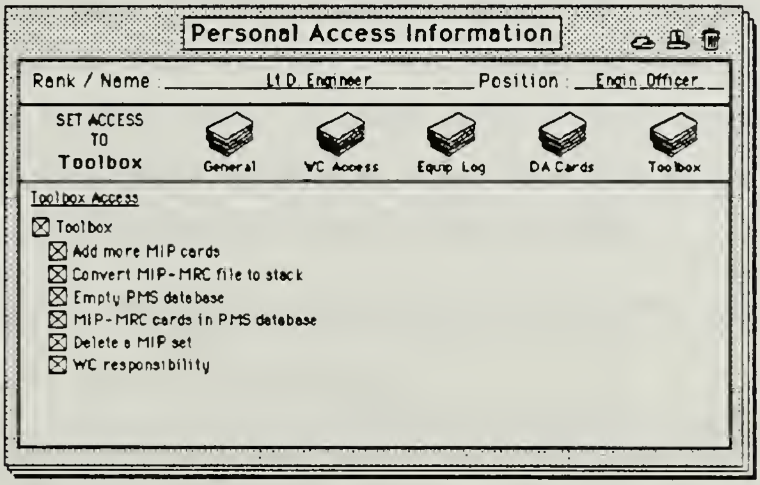
PMS Menus

☒ Check job
  ☒ Done
  ☒ Cancel check
  ☐ Cancel PMS job
☒ Notes
  ☒ Fill
  ☒ See
  ☐ See all
☒ Equipment log
  ☒ Go to log
  ☐ Add job to log
☐ Personel
  ☐ Assign job
  ☐ Rmv person
☒ Go
  ☒ MRC card ☒ MIP card
☒ Comment

Repair Maintenance

☐ Arrange repair job
  ☐ Add a job
  ☐ Delete a job
  ☐ Transfer a job
☒ Check repair job
  ☒ Done
  ☒ Cancel check
  ☐ Cancel repair job

Toolbox This button presents to the SuperUser, the Toolbox access menu. It controls access to the Toolbox menu of the driver. In general , only the SuperUser has access to this functions.



Rank / Name : Lt D. Engineer
Position : Engin. Officer

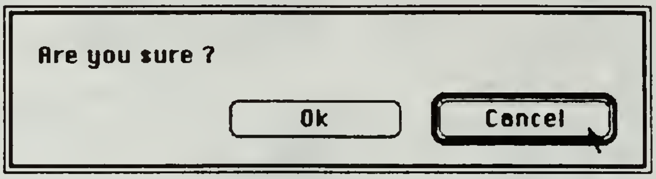
Toolbox Access

☒ Toolbox
  ☒ Add more MIP cards
  ☒ Convert MIP-MRC file to stack
  ☒ Empty PMS database
  ☒ MIP-MRC cards in PMS database
  ☒ Delete a MIP set
  ☒ WC responsibility

This button is an overloaded button. Its function is to store the current card in the password database if it is new or to return the SuperUser to the PMS driver if it is a reviewed one.

This button produces a hard copy of the five categories for review purposes.

🗑 This button deletes the current card. When pressed, the user has the option of either cancelling the operation or proceeding by selecting the appropriate button.


Are you sure ?

After the user is through with this operation, he is automatically returned to the PMS driver's menus.

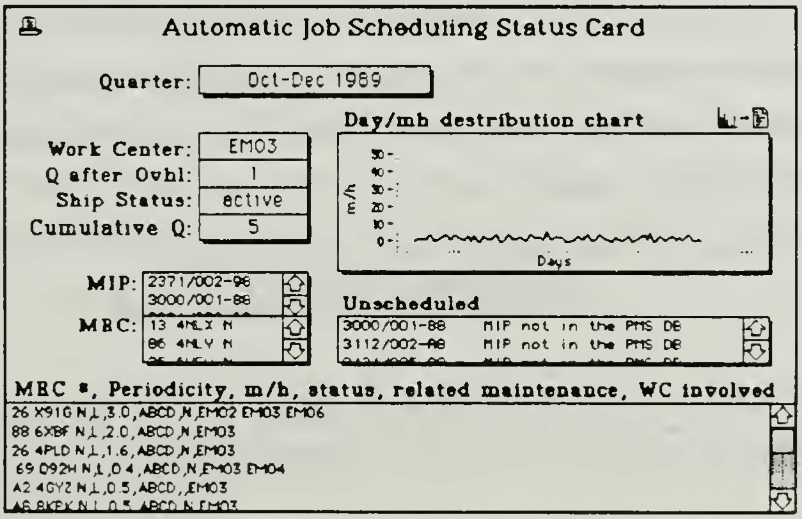
The Daily Activity stack

If the PMS driver stack can be compared to the bridge of a ship, the Daily activity stack is the engineering control room. This is where most of action takes place.

A Daily activity stack is automatically generated every time anew quarter is scheduled. The stack consists of two kinds of cards. The first ,the "Automatic job Scheduling Status Card", holds information about the job scheduling done for this quarter. The second, the actual Daily activity cards, stores information about the jobs which must be done every day. In depth description of each of these cards and their functions is given below.

The "Automatic Job Scheduling Status Card"

For every work center, one card of this type is generated and set at the beginning of each quarter. This card exists to help the division officer evaluate the system, according to the information found here, so that corrective action can be taken to improve the overall performance of the PMS system . Everything that went wrong during the scheduling process is reported here. A graph gives a visual feedback showing the distribution of the jobs throughout the quarter. Also, jobs that were not scheduled are reported along with the reason automatic scheduling has failed.


Automatic Job Scheduling Status Card
Day/mh destribution chart
m/h
50
40
30
20
10
0
Days
Unscheduled
MBC *, Periodicity, m/h, status, related maintenance, WC involved

Quarter: Oct-Dec 1989

Work Center: EM03

Q after Ovhl: 1

Ship Status: active

Cumulative Q: 5

MIP:
2371/002-98
3000/001-88

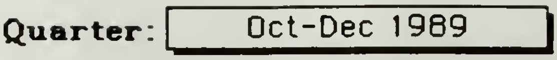
MBC:
13 4MLX N
86 4MLY N

Unscheduled:
3000/001-88 MIP not in the PMS DB
3112/002-88 MIP not in the PMS DB

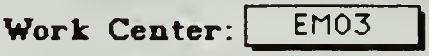
26 X91G N⊥,3.0,ABCD,N,EM02 EM03 EM06
88 6XBF N⊥,2.0,ABCD,N,EM03
26 4PLD N⊥,1.6,ABCD,N,EM03
69 092H N⊥,0.4,ABCD,N,EM03 EM04
A2 4GYZ N⊥,0.5,ABCD,,EM03
A6 8KFK N⊥,0.5,ABCD,N,EM03

The fields

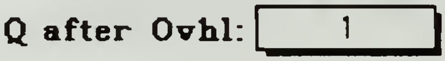
- *The "Quarter" field:* shows the name of the quarter scheduled.


Quarter: Oct-Dec 1989

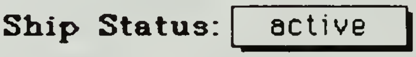
- *The "Work Center" field:* shows the name of the work center to which the card belongs.


Work Center: EM03

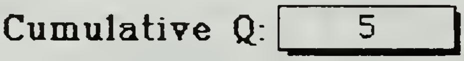
- *The "Quarter after Overhaul" field:* shows the number of the quarter since the last major ship's overhaul was.


Q after Ovhl: 1

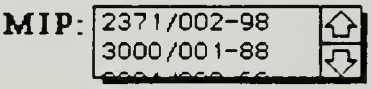
- *The "Ship Status" field:* Two values , "Active" or "Inactive", represent the current ship status. The reason this field is here is because jobs are to be done according to the ship's status.


Ship Status: active

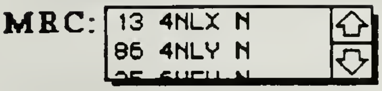
- *The "Cumulative quarter" field:* The cumulative quarter is a number that shows the number of quarters since the PMS system was installed. The purpose of this number is for scheduling jobs that do not belong in a quarter, such as six-month and annual jobs. By using the mod function, the system can decide which jobs, with a periodicity greater than "quarterly", should be scheduled and which should be skipped. This way the system totally eliminates the need of "long term" scheduling and is able to directly produce the daily activity cards.


Cumulative Q: 5

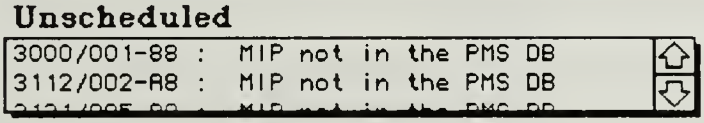
- *The "MIP" field:* In this field all MIP cards for the particular work center are listed.


MIP:
2371/002-98
3000/001-88

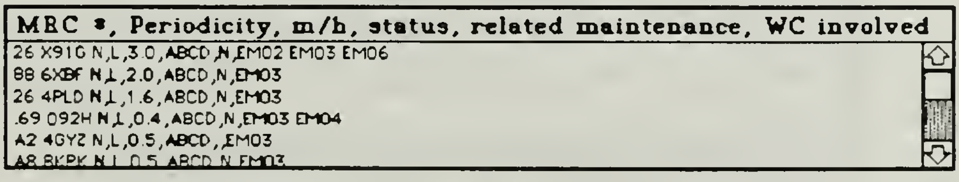
- *The "MRC" field:* In this field all MRC cards for the particular work center are listed.


MRC:
13 4NLX N
86 4NLY N

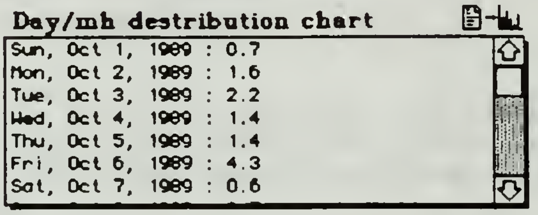
- *The "Unscheduled" field:* This field is one of the most important for the whole system, because it gives feedback information to both the PMS coordinator and the division officer. It shows which jobs could not be automatically scheduled along with the reason the system failed to do so. Some jobs, like "situation requirement" jobs, cannot be scheduled automatically and the division officer should schedule them manually. In this field, MIP and MRC cards not in the PMS database are reported so that the PMS coordinator can add them. The system also reports jobs, with a periodicity longer than "quarterly", if the desired quarter to be scheduled is not set. This function is explained in detail in the description of the "data" field of the MRC cards in the appropriate section of this manual.


Unscheduled
3000/001-88 : MIP not in the PMS DB
3112/002-A8 : MIP not in the PMS DB

- *The "MRC#, Periodicity, m/h, status,related maintenance,WC involved" field:* After the system searches the PMS database and finds out whether the MRC cards exist or not, it sets up information in this field relevant to each MRC card it is going to schedule. Then it deletes this job, from the other involved work centers so that the same job is scheduled on the same day for all the work centers. After this field is set for all work centers, the system takes the MRC jobs, one by one, and performs the scheduling.


MRC #, Periodicity, m/h, status, related maintenance, WC involved
26 X91G N,L,3.0,ABCD,N,EM02 EM03 EM06
88 6XBF N,L,2.0,ABCD,N,EM03
26 4PLD N,L,1.6,ABCD,N,EM03
.69 092H N,L,0.4,ABCD,N,EM03 EM04
A2 4GYZ N,L,0.5,ABCD,,EM03
A8 8KPK N,L,0.5,ABCD,N,EM03

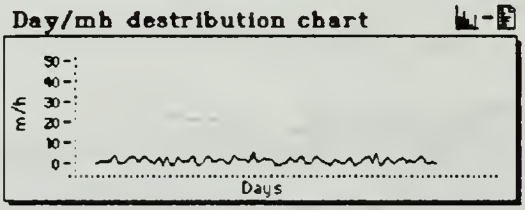
• *The "Day/mh distribution chart" field:* This field is presented in two ways according to what information the user wishes to extract from it. The first is the list view, where the user can see each day, along with the man/hours that PMS jobs have loaded it. An example of this view is shown below.



Day/mh destribution chart

```
Sun, Oct 1, 1989 : 0.7
Mon, Oct 2, 1989 : 1.6
Tue, Oct 3, 1989 : 2.2
Wed, Oct 4, 1989 : 1.4
Thu, Oct 5, 1989 : 1.4
Fri, Oct 6, 1989 : 4.3
Sat, Oct 7, 1989 : 0.6
```

The second is the graph view. The same information is presented in a graphic form allowing the user an immediate view of whether or not the scheduling has been done an efficiently. The smoother the curve, the smaller the peaks(the difference between minimum and maximum in the m/h scale), the better the scheduling has been done. The graphic form, for the data above, follows.

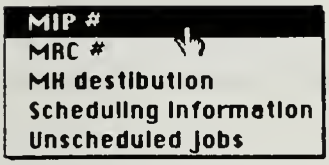


The buttons

📊→📄 This button changes the view of the days/mh distribution field from graph to list.

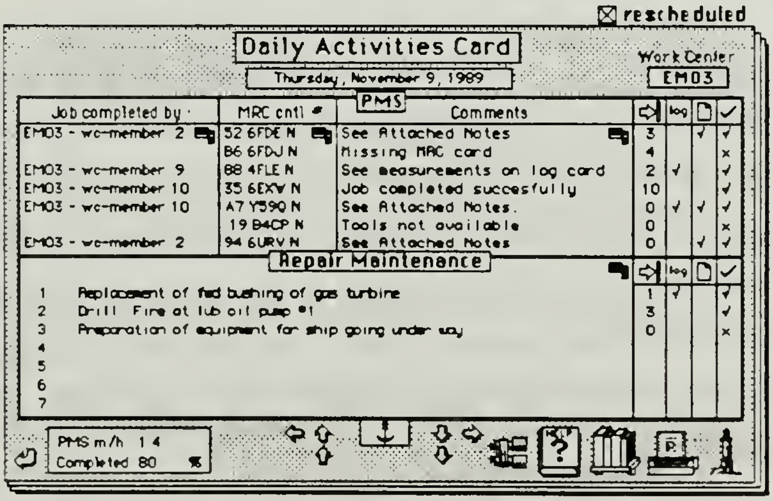
📄→📊 This button changes the view of the days/mh distribution field from list to graph.

🖶 When the user presses and holds down this button, they are presented with the pop-up menu shown below. It allows the user to choose from which field a printed copy is to be generated.



The "Daily Activities Card"

The "Daily Activity" cards are located after the "Automatic Job Scheduling Status" cards in the Daily Activity stack. One card is generated for each work center for each day of the quarter. This card serves two purposes. The first as an assignment sheet, where the jobs, both for preventive and repair maintenance, that a work center has to do are found. The second is for storage for later reference and reports generation. The user proceeds directly from the work center menus of the PMS driver to the DA card of today. If this is the first time the card is opened for the day, automatic rescheduling of uncompleted jobs from the previous day takes place. In a typical day, the person responsible for assigning the PMS jobs, assigns them using the "Personnel" function as described below. The division officer and the head of the work center, using the PMS man/hours for the day as a reference, can assign repair maintenance jobs to be done for the particular day. In case they feel that a job can wait until the next day , they can leave the job unassigned and the job will automatically be transferred to the next day. The card is divided to two parts. The top part for Preventive maintenance and the bottom part for the repair maintenance . Those two parts are described further on..


☒ rescheduled

Daily Activities Card

Thursday, November 9, 1989

Work Center EM03

PMS

Job completed by	MRC cntl #	Comments	⇨	log	🗋	✓
EM03 - wc-member 2	52 6FDE N	See Attached Notes	3		✓	✓
	B6 6FDJ N	Missing MRC card	4			x
EM03 - wc-member 9	88 4FLE N	See measurements on log card	2	✓		✓
EM03 - wc-member 10	35 6EXV N	Job completed succesfully	10			✓
EM03 - wc-member 10	A7 Y590 N	See Attached Notes.	0	✓	✓	✓
	19 B4CP N	Tools not available	0			x
EM03 - wc-member 2	94 6URV N	See Attached Notes	0		✓	✓

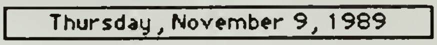
Repair Maintenance

		⇨	log	🗋	✓
1	Replacement of fwd bushing of gas turbine	1	✓		✓
2	Drill Fire at lub oil pump #1	3			✓
3	Preparation of equipment for ship going under way	0			x
4					
5					
6					
7					

PMS m/h 1 4
Completed 80 %

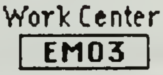
The fields

- *The "Date" field:* This field contains the date of the card being are looking at.



Thursday, November 9, 1989

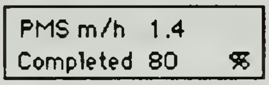
- *The "Work Center" field:* This field contains the name of the work center the card belongs to.



Work Center

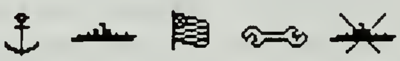
EM03

- *The "PMS m/h" and "Completed" field:* This field consists of two parts. The top, labeled "PMS m/h", contains the total man/hours that are needed to complete all the PMS jobs for the day. The second, labeled "Completed", contains the percentage of PMS jobs that are completed and the m/h. This field is updated every time a change takes place in the status of the PMS jobs. This field is also used to generate the report graphs.



PMS m/h 1.4
Completed 80 %

- *The "Status" field:* This field contains icons instead of characters. The icons are the same ones described in the description of the "Planning" stack and are set by the SuperUser during the scheduling process.



- *The "rescheduled" field:* This technically speaking is a button, but since it acts like a field, it is described here. This field is X-ed out when rescheduling has been done for the card we are looking at. If we look at tomorrows card, this field is empty. It serves both as a reference and as an indication to the system of how many days must be rescheduled. This means that if no one opens the DA card for a particular work center for several days, the first time today's card is opened, rescheduling will take place for the previous days that were not rescheduled. The jobs from all those days will be transferred forward.

☒ rescheduled

The buttons

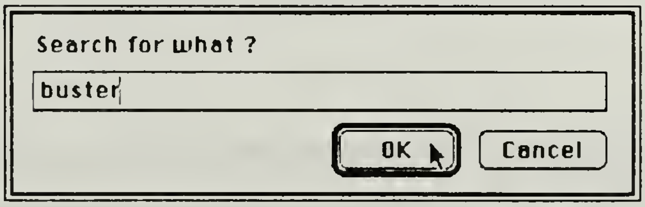
⇨ This button shows the user the next day's DA card. If this is the last card of a quarter, the user may not go to another work center.

⇦ This button shows the user the previous day's DA card. If this is the first card of a quarter, the user may not go to another work center.

⇧
⇧ This is a set of two buttons at the bottom of the card. They serve the purpose of scrolling the fields of the card allowing users to see contents of the field out of view. The top arrow button scrolls the top set of fields up, while the bottom arrow button scrolls the bottom set of fields up.

⇩
⇩ This set of buttons function exactly as above , except they scroll the corresponding fields down.

This button invokes the search function of the system. Searches are global, but only relevant for the work center from which called. The user is prompted to enter the search string in the following dialog box . The string can be a whole word or part of a word.



Search for what ?

buster

If the desired string is not found,the user is informed.

String not found

The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

This button is a link to the Technical library. The current card is temporarily stored away and the user is presented with the Technical library stack.

The print button prints a copy of the current card. The format of the DA card is printed slightly different from that viewed on screen. This is so the fields can be expanded to their full length. This function can be invoked directly from the PMS driver, to print the current day's card. The printed copy can be used for reference during the day's job assignment and as a report for the day.

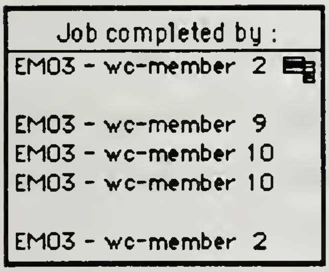
This button returns the user to the menu card of the PMS driver.

This button allows the user to exit the PMS system and return to the log-in card of the driver.

Preventive Maintenance (top part)

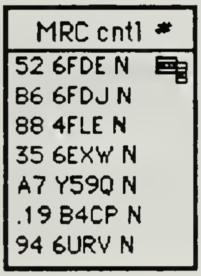
The fields

• *The "Job completed by:" field:* In this field the division officer or the work center supervisor assigns the name of the person or persons who are going to perform the PMS job. The individual(s) is responsible for ensuring that the job is completed and for assigning notes and comments to the job. If a job has no name in this field, it means that the division officer or work center supervisor, desires that the job be carried forward to the next day.



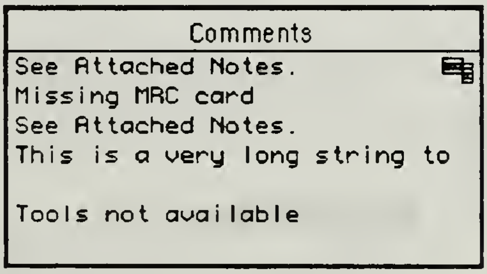
Job completed by :
EM03 - wc-member 2
EM03 - wc-member 9
EM03 - wc-member 10
EM03 - wc-member 10
EM03 - wc-member 2

• *The "MRC cntl #" field:* In this field, the system automatically enters the PMS jobs to be performed . Jobs not completed during the previous days are moved forward and placed following the jobs that were originally scheduled for the day.



MRC cntl #
52 6FDE N
B6 6FDJ N
88 4FLE N
35 6EXW N
A7 Y59Q N
.19 B4CP N
94 6URV N

• *The "Comments" field:* This field is where short comments are placed for a particular job. The person making the comments, can either pick one from the list as described in the menus' description or write their own.



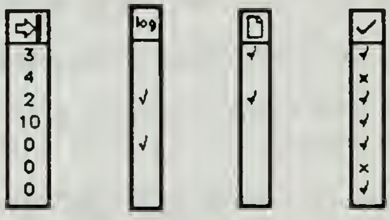
Comments
See Attached Notes.
Missing MRC card
See Attached Notes.
This is a very long string to
Tools not available

• *The " ⇥ " field:* In this field we can find an integer number. This number indicates how many days a job has been delayed from its original scheduling. This number, placed here by the system, is used by the division officer or work center supervisor to assign priority of a job over a another job that has been delayed for fewer days.

• *The " log " field:* In this field we find a "✓" mark if the job has been transferred into the equipment's log page.

- *The "🗋" field:* A "✓" mark is found in this field if notes associated with this job exist.
- *The "✓" field:* In this field a "✓" mark indicates that the job has been completed successfully. An "x" mark indicates that the job has been canceled.

An instance of the above four mentioned fields follows.

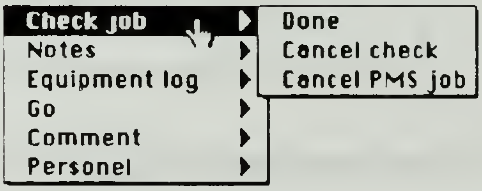


Note: The ☰ symbol in any of the above fields indicates that a pop-up menu is associated with this field, and the user can click in the field to show the different options available to them according to their access.

The menus

When the user holds down the mouse button in a field with the ☰ sign, they are presented with the following menus. The menus take information from the item they are popped up from. The actions taken apply to the particular selected item.

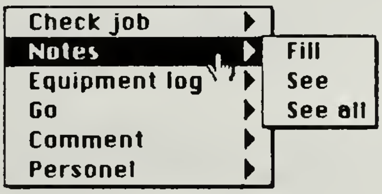
The "Check job" choice: When this item is selected, the user can do one of the operations on the job whose line they invoked the menu from.



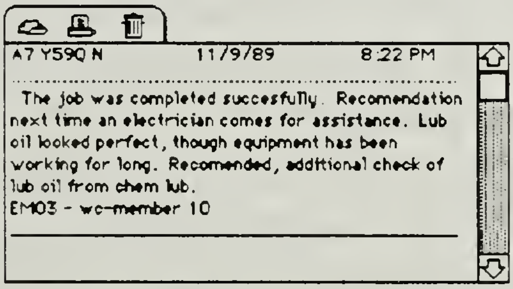
- *"Done":* This item is available to everyone that has access to the system. By selecting it, a "✓" mark is assigned in the ✓ field next to the job selected.
- *"Cancel check":* This item is available to everyone that has access to the system. By selecting it, a job that has been checked "completed" gets unchecked. This operation cancels the previous one.

- *"Cancel PMS job":* This item is available only to the division officer, PMS coordinator and work center supervisors. This function is used to cancel a PMS job either because it is already scheduled or because the person who has authority to cancel it thinks that it no longer applies to a day or to the ship's current situation. When selected, an "x" mark is assigned to the field next to this job. It is advisable that a relevant note or short comment be made explaining the reason the job was canceled.

The "Notes" choice: When this item is selected, the user is presented with the three item sub-menu shown below.



- *"Fill":* When this item is selected, the system either creates a new "note page" if one has not already been created or opens the existing one if notes have been previously assigned to this job .

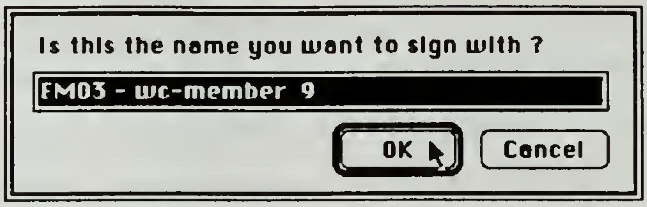


A7 Y59Q N 11/9/89 8:22 PM

The job was completed succesfully. Recomendation next time an electrician comes for assistance. Lub oil looked perfect, though equipment has been working for long. Recomended, additional check of lub oil from chem lub.
EM03 - wc-member 10

On opening a "note page", the system automatically assigns the MRC #, the date and the time . After the user types in notes or comments about the job performed, three options, activated by the corresponding icons at the top flap of the "note page", are available.

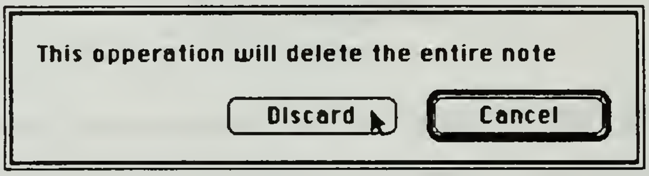
✍ This button stores the note after signing . When pressed, a dialog pops up asking the user for the name he wishes to sign with. As a default, the name of the person responsible for the PMS job is given. Further checking can be forced , if needed, by modification of the program. For the time being, it is not considered necessary.


Is this the name you want to sign with ?

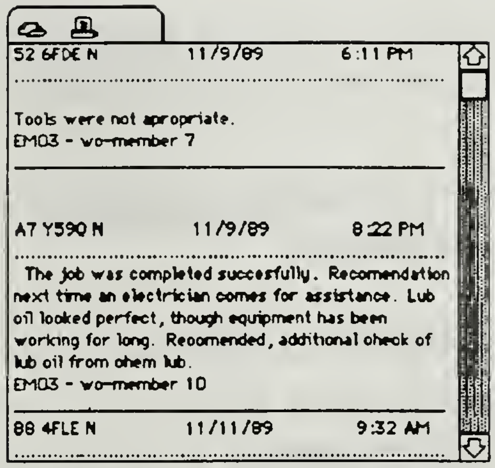
FM03 - wc-member 9

🖶 This button generates a printed copy of the "note page".

🗑 This button discards the whole note if the user thinks that it is no longer needed. Prior to throwing the note away, the following dialog box warns the user.


This opperation will delete the entire note

- *"See":* When this item is selected, the user can review,but not alter, any note. This function protects the system from unauthorized alteration of information but still allows the reviewing of notes. Buttons with the same functions as above are provided, but without the signature capability.
- *"See all":* This item, generates a list of all the notes for the particular day. It is provided as a way for the division officer or department head to view all the jobs and the notes associated with them for their work centers. The options are the same as above, but without the "delete note" button, because the information in this field is generated from scratch every time, it is requested.



52 6FDE N 11/9/89 6:11 PM

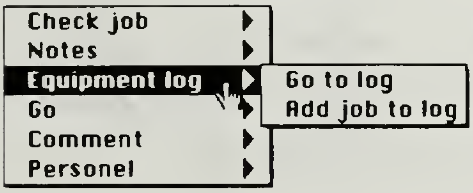
Tools were not apropriate.
EM03 - wo-member 7

A7 Y590 N 11/9/89 8:22 PM

The job was completed succesfully. Recomendation next time an electrician comes for assistance. Lub oil looked perfect, though equipment has been working for long. Recomended, additional check of lub oil from chem lub.
EM03 - wo-member 10

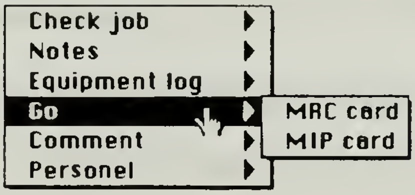
88 4FLE N 11/11/89 9:32 AM

The "Equipment log" choice: When this item is selected, the following menu appears.



- *"Go to log":* This selection shows the corresponding log card, provided a card exists and the link has been set up, as described in the "PMS stack" description of this manual, to the particular piece of equipment to which this job belongs.
- *"Add job to log":* When this selection is chosen, the job and today's date, are transferred to the appropriate card of the log stack. This function should be used with care to avoid transferring jobs that are not important or excluding vital maintenance jobs from being transferred. Future releases of the system should perform this transfer automatically, when a job is completed according to a preset look up table.

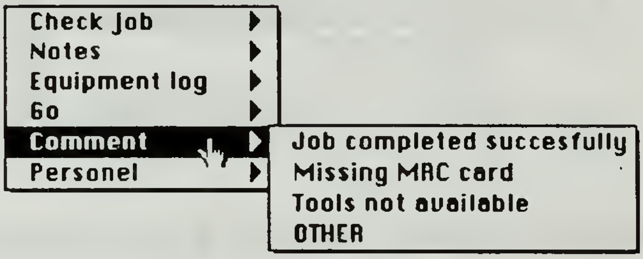
The "Go" choice: This item has two sub-choices to select from.



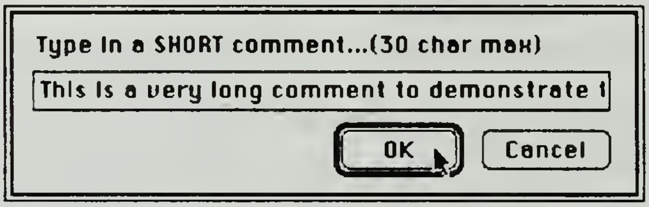
- *"MRC card":* By selecting this item, the user can see the MRC card, from which the function was invoked.

- *"MIP card":* This function shows the user the corresponding MIP card of the piece of equipment to which the MRC he clicked on refers.

The "Comment" choice: This item allows the user to add a short comment next to the MRC job, about its implementation.

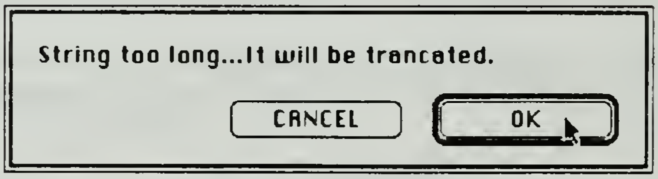


The user has the choice of either selecting one of the predefined items from the list or adding their own by selecting the "OTHER" choice. If the "OTHER" choice is selected, the following dialog appears, asking to enter a comment up to thirty characters long.

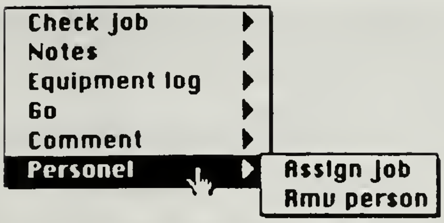

Type In a SHORT comment...(30 char max)

This Is a very long comment to demonstrate t

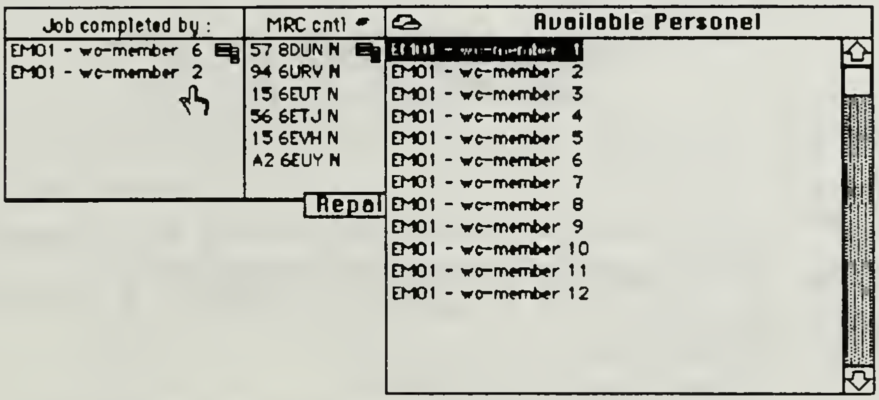
In case the string entered is longer than the thirty characters allowed, the user is presented with another dialog, requesting whether they want the comment entered to be truncated to fit.


String too long...It will be trancated.

The "Personnel" choice: This choice is available to division officers, work center heads and the SuperUser . It has two functions.



- *"Assign job":* This function allows the user to assign the MRC jobs of the day to individuals. When selected, a list of the available personnel for the particular work center is presented.



Job completed by :	MRC cntl	Available Personel
EM01 - wo-member 6	57 8DUN N	EM01 - wc-member 1
EM01 - wc-member 2	94 6URV N	EM01 - wc-member 2
	15 6EUT N	EM01 - wc-member 3
	56 6ETJ N	EM01 - wc-member 4
	15 6EVH N	EM01 - wc-member 5
	A2 6EUY N	EM01 - wc-member 6
		EM01 - wo-member 7
		EM01 - wc-member 8
		EM01 - wc-member 9
		EM01 - wc-member 10
		EM01 - wo-member 11
		EM01 - wo-member 12

The user, first clicks on a person's name in this field to select him. The name selected becomes highlighted. After the name selection, next click on the line of the mrc job to assign this person is transferred their name to this line. The whole assignment process takes only few seconds. A persons name can be overwritten by another selection if the user changes his mind. To remove the name list from the screen, the ☁ button must be pressed.

- *"Rmv person":* This function removes the name assigned from the line from which it was invoked.

The Repair or Corrective Maintenance (bottom part)

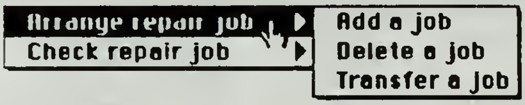
The fields

The fields in this part are the same as the described above. The only difference one is the "Repair Maintenance" field which is the one described below.

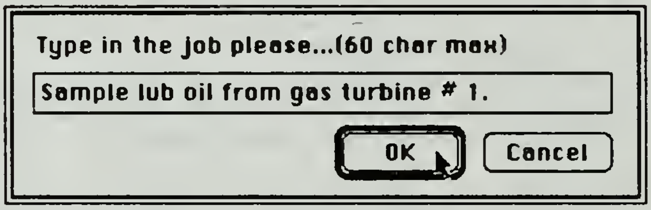
- *"Repair Maintenance":* This field is filled in manually by the division officer or the work center head. As the icon on the top right corner indicates, there is a pop up menu assigned to it. Here they can add repair maintenance jobs for the day, or training subjects for the particular work center. To do this, the user consults the total m/h number for the day in order to avoid overloading it, and then presses the mouse button within the field. The following pop up menu appears.

The menu

The "Arrange repair job" choice: This choice has three sub-choices.

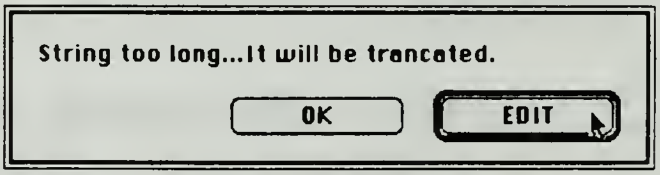


- *"Add a job":* This choice allows the user to add a new repair job in the repair maintenance list. When selected a dialog box is presented, to allow the user type in the repair job's description.

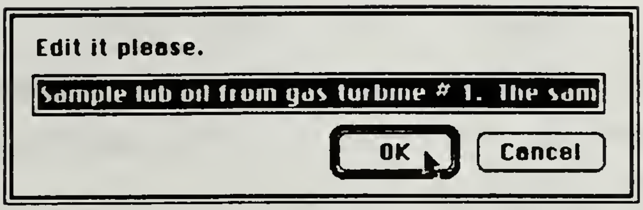

Type in the job please...(60 char max)

Sample lub oil from gas turbine # 1.

The repair job must have a maximum length of sixty characters. In case the string typed in is longer than that, the user is prompted with another dialog box that allows for either editing the existing string by the user or allowing the system to truncate it to the appropriate length.


String too long...It will be trancated.

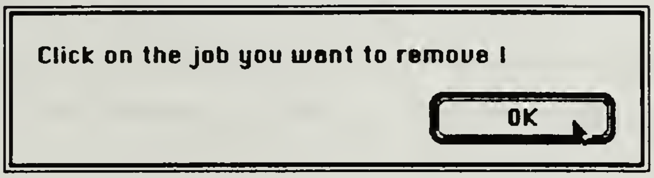
If the user selects to edit it, the string is returned, ready for editing.



Edit it please.

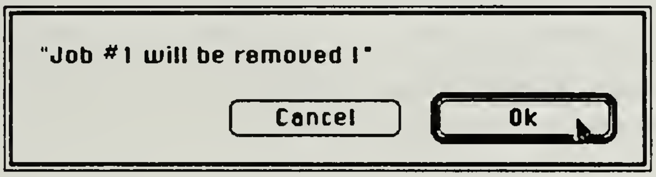
Sample lub oil from gas turbine # 1. The sam

- *"Delete a job":* This choice allows the user to delete a job previously entered. The following dialog box appears.



Click on the job you want to remove !

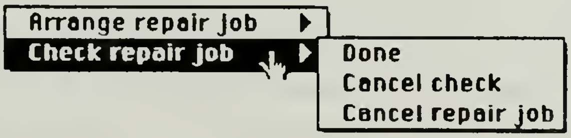
A message asks to verify that the user has selected the correct job. If "OK" is selected, the job gets deleted.



"Job #1 will be removed !"

- *"Transfer a job":* This choice allows the user to transfer the selected job to another day.

The "Check repair job" choice: This choice has three sub-choices . This function is available to everyone that can confirm that a job has been completed.



- *"Done":* This selection checks a job as completed.
- *"Cancel check":* This selection removes a check accidentally entered.
- *"Cancel repair job":* This function, available only to those persons having the authority of entering jobs, gives the option of cancelling a repair job, instead of removing it.

Hint: The better this stack is maintained and the more precise and faltless the filling of information is, the better the PMS system is going to work for the individual ship. Old quarters can be kept aside for reference and history purposes. A quarter can fit in an 800 K floppy disk in the ship's library in order to form a digital record of all the PMS done during the ship's life. The DA stacks have to be compressed though, using the "Stuffit" program provided along.

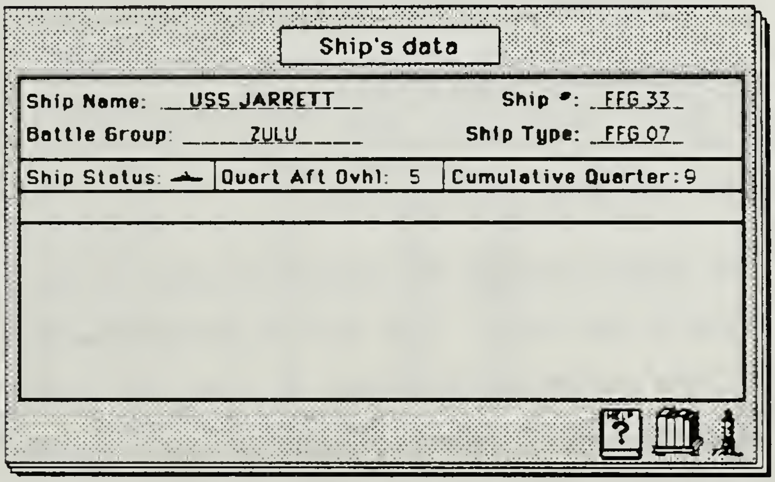
The Ship data stack

General description

The Ship data stack contains general information about the ship. For the present, it consists of three kinds of cards. The capability is to add more is present if necessary in the future.

The "Ship's Data" card

This card contains general information about the ship that is used through out the system. Explanation of the individual fields is not included because they actually describe themselves. The buttons of this card are the same as used throughout the system.



Ship Name: USS JARRETT

Ship #: FFG 33

Battle Group: ZULU

Ship Type: FFG 07

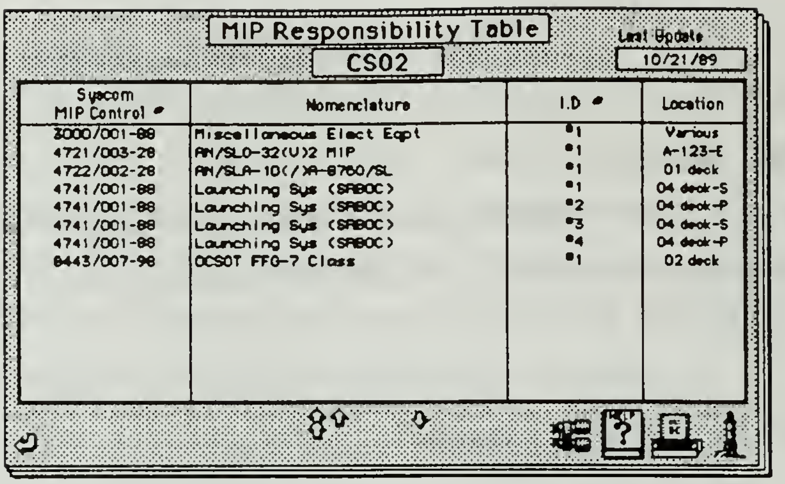
Ship Status:

Quart Aft Ovhl: 5

Cumulative Quarter: 9

The "MIP Responsibility Table" card

This type of card contains a list of all the MIP cards a work center is responsible for. Multiple entries of the same MIP number mean that the particular work center is responsible for more than one piece of equipment of this type. For example, if the work center whose card is shown below is responsible for four RBOC's, an entry for each should be present so as to schedule the same job four times, once for each unit.



MIP Responsibility Table

CS02

Last Update
10/21/89

Syscom MIP Control #	Nomenclature	I.D #	Location
3000/001-88	Miscellaneous Elect Eqpt	#1	Various
4721/003-28	AN/SLQ-32(V)2 MIP	#1	A-123-E
4722/002-28	AN/SLA-10(/)A-8760/SL	#1	01 deck
4741/001-88	Launching Sys (SRBOC)	#1	04 deck-S
4741/001-88	Launching Sys (SRBOC)	#2	04 deck-P
4741/001-88	Launching Sys (SRBOC)	#3	04 deck-S
4741/001-88	Launching Sys (SRBOC)	#4	04 deck-P
8443/007-98	OCSOT FFG-7 Class	#1	02 deck

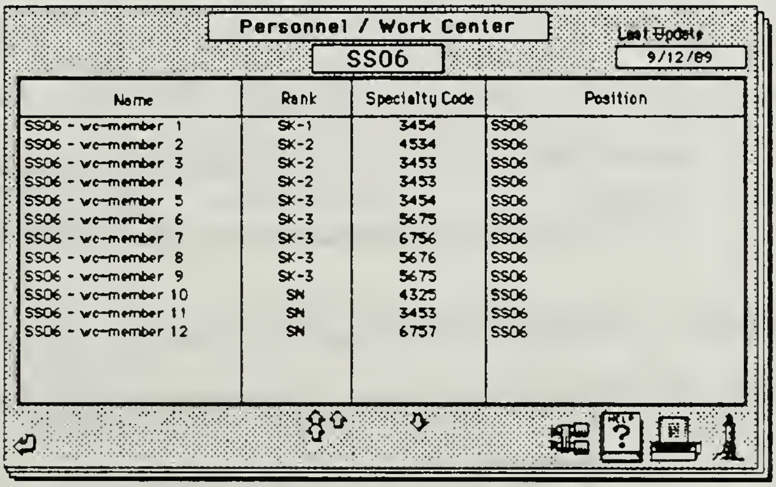
The buttons

⇮	Gets the user to the top of the page.
⇧	Gets the user one screen up.
⇩	Gets the user one screen down.
	This button invokes the search function of the system. Searches are global in the stack. The user is prompted with a dialog box, similar to that the previously described, to enter the search string. The string can be a whole word or part of a word.
HELP ?	The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.
🖶	The print button prints the contents of this card.
↩	This button returns the user to the first card of the ship's data stack.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "Personnel/Work Center" card

In this card, the names, rank, specialty code, and position of each person involved with the PMS system are entered. There exists one card for each work center. The data from this card is used when the "Personnel" function is invoked from the PMS Daily activity card. When the personnel function for Argos becomes available, data about the people assigned to every work center should be sent to this card in order to allow the division officer or the person who has the responsibility of assigning the day's jobs to have an updated list of the available personnel for a particular day. Right now all the personnel assigned to a particular work centered are presented here.



Personnel / Work Center

SS06

Last Update 9/12/89

Name	Rank	Specialty Code	Position
SS06 - wc-member 1	SK-1	3454	SS06
SS06 - wc-member 2	SK-2	4534	SS06
SS06 - wc-member 3	SK-2	3453	SS06
SS06 - wc-member 4	SK-2	3453	SS06
SS06 - wc-member 5	SK-3	3454	SS06
SS06 - wc-member 6	SK-3	5675	SS06
SS06 - wc-member 7	SK-3	6756	SS06
SS06 - wc-member 8	SK-3	5676	SS06
SS06 - wc-member 9	SK-3	5675	SS06
SS06 - wc-member 10	SN	4325	SS06
SS06 - wc-member 11	SN	3453	SS06
SS06 - wc-member 12	SN	6757	SS06

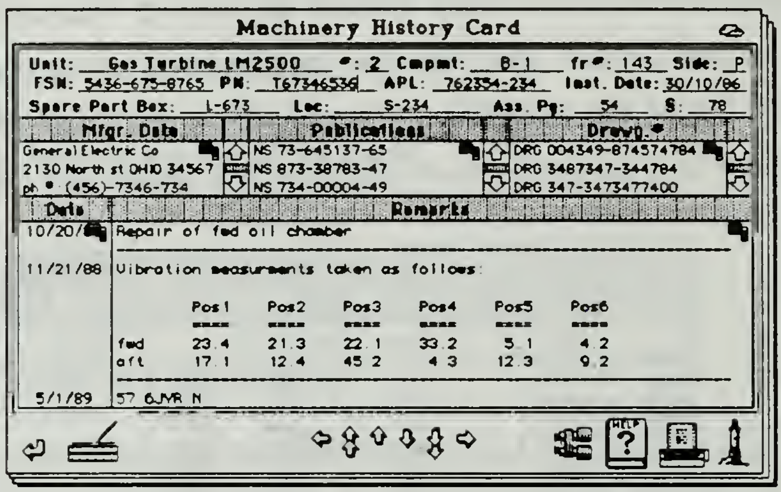
The buttons

The buttons described in the previous section are available in these cards too.

The Log stack

General description

The "Log" stack is available to all hands, although not everyone has the same access privileges. This stack is where we keep the history of each piece of equipment on board the ship. When a particular unit is removed from a ship, to be installed to another, or to be kept as a reserve unit, both a printed and a digital copy of this card, should follow it. The card is maintained for the life of the unit. When the unit is thrown away, information from it can be used to judge the future purchase of similar equipment.



Machinery History Card

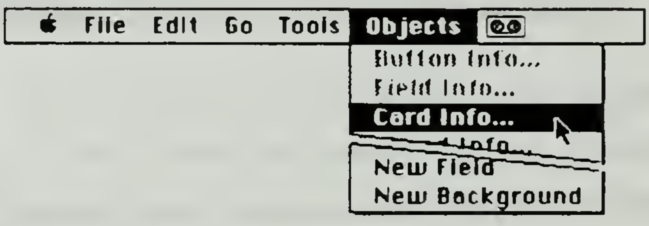
Unit: Gas Turbine LM2500 #: 2 Cmpmt: B-1 fr#: 143 Side: P
FSN: 5436-675-8765 PN: T67346536 APL: 762354-234 Inst. Date: 30/10/86
Spare Part Box: L-673 Loc: S-234 Ass. Pg: 54 $: 78

Mfgr. Data	Publications	Drawg. #
General Electric Co	NS 73-645137-65	DRG 004349-874574784
2130 North st OHIO 34567	NS 873-38783-47	DRG 3487347-344784
ph # (456)-7346-734	NS 734-00004-49	DRG 347-3473477400

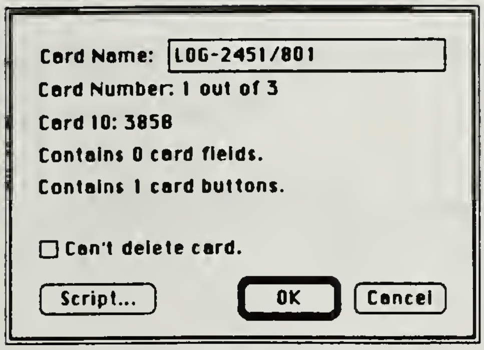
Date	Remarks
10/20/	Repair of fwd oil chamber
11/21/88	Vibration measurments taken as follows:
5/1/89	57 6JVR N

	Pos1	Pos2	Pos3	Pos4	Pos5	Pos6
fwd	23.4	21.3	22.1	33.2	5.1	4.2
aft	17.1	12.4	45.2	4.3	12.3	9.2

When a new card is created, is very important to set the name of the card correctly. To do this, we open the "Objects" menu from the menu bar, and select "Card info...".



When the dialog box appears, fill in for "Card Name:" the word "LOG" a "-" and the first eight digits of the corresponding to the equipment MIP card, as shown in the example. Then type <return> or click in the "OK" box, to set it. This will also set the connection to the corresponding MIP/MRC cards.


Card Name: LOG-2451/801
Card Number: 1 out of 3
Card ID: 3858
Contains 0 card fields.
Contains 1 card buttons.

☐ Can't delete card.

The fields

Choices on the fields described below are according to the user's access clearance.

• *The "Unit" field:* This field contains the name of the unit the card belongs to. In the general case, this is the same as the unit name assigned to the appropriate MIP card in the corresponding field.

Unit: ....

• *The "#" field:* This field distinguishes between units with the same name. For example, if a ship has two LM2500 gas turbines, there should be one card for each, with "#1" in this field for the starboard, and "#2" for the port one.

#: ....

• *The "Cmpmt" field:* In this field, the compartment of the ship where the equipment is located is assigned.

Cmpmt: ....

• *The "fr#" field:* Shows the approximate frame number where the unit is located. This information further identifies the unit and it can also be used on reports that are generated about this piece of equipment.

fr#: ....

• *The "Side" field:* In this field, a "P" or an "S" is assigned for "port" and "starboard" side correspondingly.

Side: ......

• *The "FSN" field:* This field contains the Federal Stock Number of the equipment.

FSN: ....

- *The "PN" field:* In this field, the Part Number of the equipment is assigned.

PN: ....

- *The "APL" field:* In this field, the Allowance Part List page for the equipment is assigned.

APL: ....

- *The "Inst. Date" field:* This field contains the date the particular unit has been installed on board the ship.

Inst. Date: ....

- *The "Spare Part Box" field:* In this field, we fill in the number of the box the spare parts for the particular equipment are stored.

Spare Part Box: ....

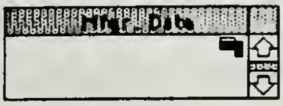
- *The "Loc" field:* In this field, we find the compartment of the ship, where the spare part box is stored.

Loc: ....

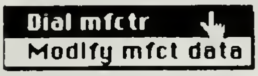
- *The "Ass. Pg" field and the " §" field:* These two fields hold information about where in the ship's enrollment list this equipment is enrolled.

Ass. Pg: .... **§:** ....

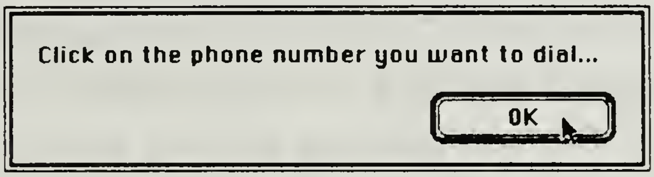
- *The "Mfgr. Data" field:* This field contains all the known information about the manufacturer of the particular unit. This is both for reference purposes, and to make correspondence with the manufacturer easier. The ▤ symbol, that appears on the right top corner of this field, indicates the existence of a pop up menu. The pop up menu in this field gives to the user two choices.



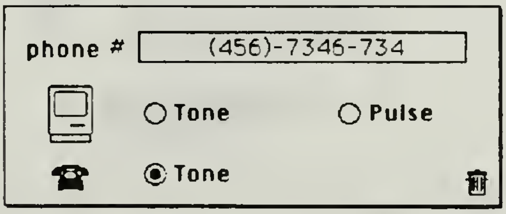
The "Dial mfctr" choice: When this choice is selected, the user can instruct the computer to dial the manufacturer electronically,provided a phone number is available.



After the mouse button is released, a dialog box asks the user to select the phone number to dial. This is set here to allow the user to specify which phone number, if multiple ones are available, to dial in order to select a fax, voice, or a data connection.


Click on the phone number you want to dial...

In order for this function to work, the line containing the phone number should have for its first five characters, the sequence "ph #:". After a number has been selected, the user is presented with the following dialog box.


phone # (456)-7346-734

○ Tone ○ Pulse

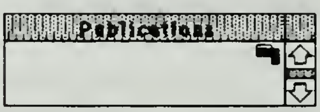
◉ Tone

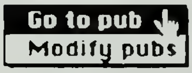
In the dialog box, the user can change the phone number and the way dialing is to be done. If a modem or a fax machine is connected to the computer, the user must specify whether the phone line is "pulse" or "tone". If no modem is available and the phone is of the "tone" variety , the user can select the appropriate button and place the phone's receiver next to the computer and let the computer generate the appropriate tones to do the dialing. To initiate the appropriate dialing process selected, click on one of the following buttons.

🖥	To dial through a modem or fax machine.
☎	To directly dial through the phone's receiver .
🗑	To cancel the dialing procedure.

The "Modify mfct data" choice: When this item is selected, the system disables the menu from this field and allows editing. When the user leaves this card to go to another one, the menu is automatically enabled again .

• *The "Publications" field:* In this field, the publications relevant to the particular piece of equipment are entered. This is for reference purposes only at this time. If in the future direct linking is needed to the appropriate library, the system lends itself for easy modification. For the present, the user has to select the library function from the bottom of the page in order to do a normal library search. The ▤ symbol that appears on the right top corner of this field indicates to the user the existence of a pop up menu. The pop up menu in this field gives to the user two choices.

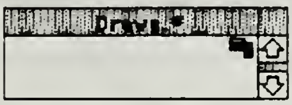


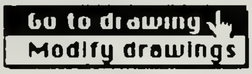


The "Go to pub" choice: This choice is provided for easy addition for direct linking to the library in the future as described above. Right now, it has the same effect as clicking on the library function at the bottom of the page.

The "Modify pubs" choice: This choice works exactly the same way as the "Modify mfct data" choice described above. It provides the user with the capability to modify the publications' list.

- *The "Drawg. #" field:* In this field all the drawing numbers for this piece of equipment are entered. This is only for reference purposes at this time. If in the future a "drawing and blue print" library in CD format are provided for the system, direct linking to the particular drawing selected can be done. The ▤ symbol that appears on the right top corner of this field indicates to the user the existence of a pop up menu. The pop up menu in this field gives to the user two choices.

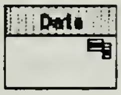


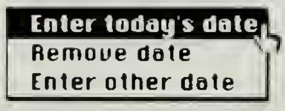


The "Go to drawing" choice: This choice will show the user the selected drawing when a "drawing" library becomes available.

The "Modify drawings" choice: This choice works exactly the same as the two above "Modify" choices do. It allows the user to modify the drawings' list.

- *The "Date" field:* In this field, the date that an entry has been made, is entered. A pop up menu is provided to let the user select one of three choices.



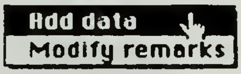


The "Enter today's date" choice: This choice automatically enters the date taken from the computer's clock into the line from which the pop up menu was invoked.

The "Remove date" choice: This choice deletes the date from the line from which the pop up menu was invoked.

The "Enter other date" choice: This choice disables the pop up menu and opens the field for editing in order to set a date, different from today's.

- *The "Remarks" field:* In this field, the user enters any remarks, comments, major repair and preventive maintenance jobs, or any measurements and test results relevant to this piece of equipment. The entries in this field should be separated with a line across the page. An option for automatic entry of this line is provided by the system and described below. A pop up menu lets the user select one of the two available operations that can be performed on this field.



The "Add data" choice: When this choice is selected, the user is presented with the last line of the "Remarks" field. The menu is disabled and any new data can be added. This choice is provided to avoid scrolling the field all the way to the bottom, since after a couple of years this field can grow considerably.

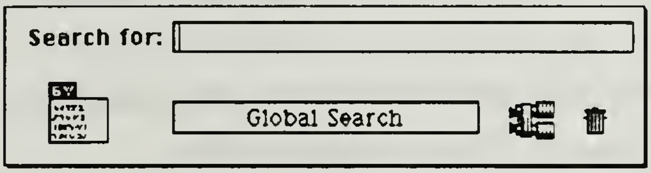
The "Modify remarks" choice: This option works the same as the previous one except that it keeps the user at the current position of the "Remarks" field.

The buttons

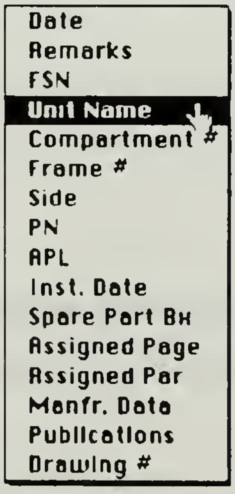
⇨	Presents the next log card.
⇦	Presents the previous month.
⇧	Gets the user one screen up in the "Remarks" field.
⇩	Gets the user one screen down in the "Remarks" field.
⇮	Gets the user to the top of the "Remarks" field.
⇟	Gets the user to the bottom of the "Remarks" field.

This button puts a line after the current line of the "Remarks" field. It is used to separate the jobs in order to make them stand better out.

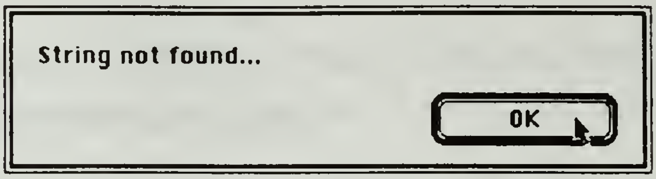
This button invokes the search function of the Log stack. When pressed, the dialog box that is shown below is presented to the user.



This dialog box is different from the ones previously shown, because it has fields and buttons of its own. The "*Search for*" field is provided so that the user can type in the string to search for. The field on the bottom is used to specify the type of search to perform. The default setting is a "Global Search". To change this and restrict the search to one specific field, press the button to invoke the pop up menu. The pop up menu, shown below, has the names of fields to search on. When the user presses the mouse button to invoke the menu, the pointer is automatically set to the "Unit Name" selection, which is the most likely the one the user is going to select.



After specifying both the search key and the field to search on, press the button or the return key, and the search begins. If the key is not found, the user is informed by the following dialog box.



String not found...

In case the user changes their mind about doing a search, press the button to get rid of the search dialog.

The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

The print button prints the current log card. At the top of every printed page, the general information about the particular piece of equipment is presented. An example of a printed page is shown in the "Auxilliary stacks" section of this manual.

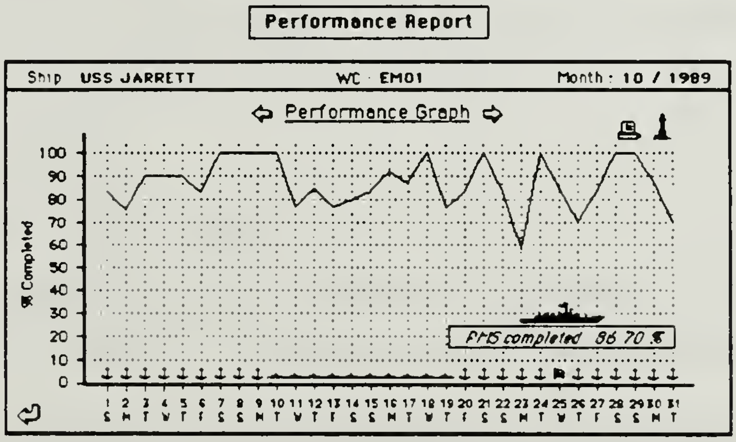
This button returns the user to the card they were on when they invoked the "log" function.

This button allows the user to exit the PMS system and return to the login card of the driver.

The Reports stack

General description

This stack is used both as a print tool and as a space where the user can go and review a report that he has requested. Every time a new report is requested, the previous one is erased. The card is a graph paper look alike card that has the days for its x-axis and the percentage of PMS jobs completed for its y-axis. Above the x-axis, a small icon represents the status of the ship for the particular day. Below the date, a letter shows the day of the week, to allow the user to better judge the PMS performance. Near the right bottom corner of the card, a field gives us the average percentage of the PMS completed for the whole month.



Performance Report

Ship USS JARRETT WC EM01 Month : 10 / 1989

PMS completed 86.70 %

The buttons

⇨ Shows the next month or the next work center if a global report for the ship has been selected.

⇦ Shows the previous month or the last month for the previous work center if a global report for the ship has been selected.

🖶 This button prints the current card.

↵ This button returns the user back to the PMS driver.

This button allows the user to exit the current card and return to the login card of the driver.

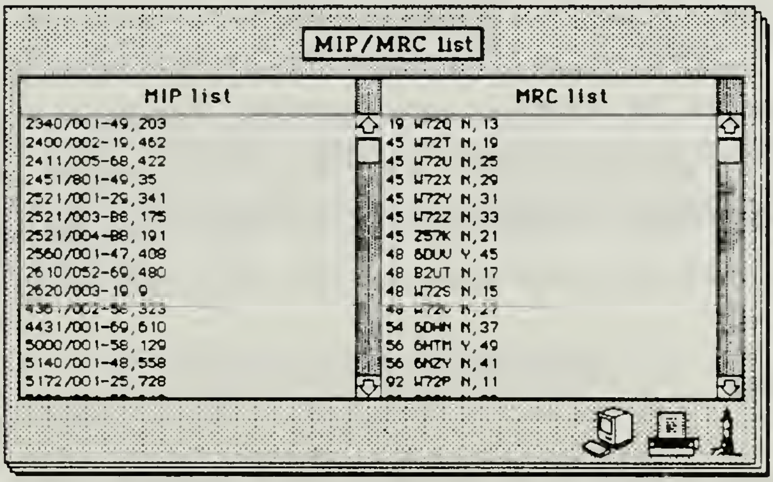
The PMS stack

General description

The PMS stack is a database of all the PMS cards in the system for a particular ship. It contains all the MIP and MRC cards, along with their tables and diagrams. One thing that the user must be aware of is that sometimes requests from this database are slow, and responses can take a couple of seconds because of its size. The system for the present contains about five thousand cards. Throughout the implementation of this database new features have been added that are not contained in the original PMS cards, but that are closely relevant to their use. It is believed that they will improve the functionality of the whole PMS system, like lists of repair parts found on the MRC cards, and the links to the "movies" stack. The hypermedia properties of the system give the capability for such additions without sacrificing the familiar view of the cards or the ease of use of the system.

The "MIP/MRC list" card

This is the first card of the PMS database. Its purpose is to allow for binary searches on the PMS database so as to improve the operating speed of the system.



MIP list

```
2340/001-49,203
2400/002-19,462
2411/005-68,422
2451/801-49,35
2521/001-29,341
2521/003-B8,175
2521/004-B8,191
2560/001-47,408
2610/052-69,480
2620/003-19,9
4361/002-56,323
4431/001-69,610
5000/001-58,129
5140/001-48,558
5172/001-25,728
```

MRC list

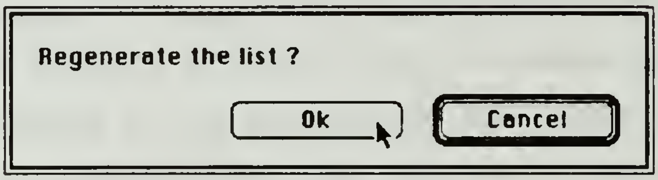
```
19 W72Q N,13
45 W72T N,19
45 W72U N,25
45 W72X N,29
45 W72Y N,31
45 W72Z N,33
45 Z57K N,21
48 6DUV Y,45
48 B2UT N,17
48 W72S N,15
48 W72V N,27
54 6DHM N,37
56 6HTM Y,49
56 6NZY N,41
92 W72P N,11
```

The fields

The only two fields found on this card, are the "MIP list" and "MRC list" fields. These fields contain a list of the corresponding cards, along with their order number in the PMS database. A binary search on this field, greatly improves the operational speed of searches.

The buttons

This button regenerates the MIP/MRC list. When pressed a dialog box asks the user to verify their choice.



Regenerate the list ?

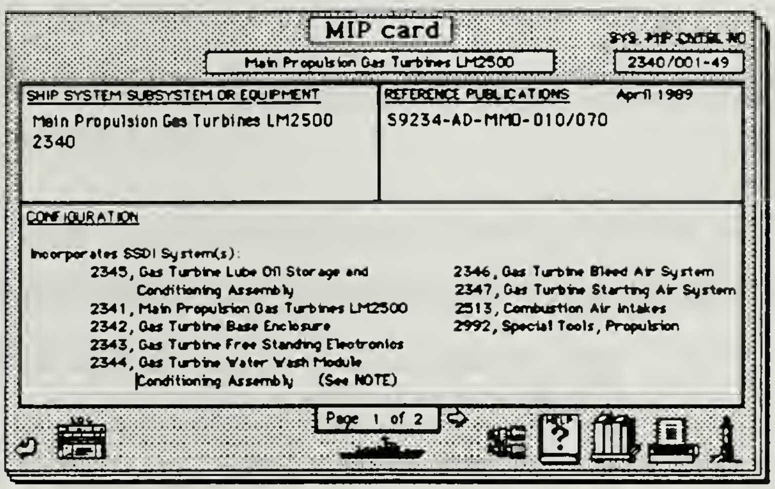
The print button invokes the "Print Report..." menu function to allow the user to obtain a hard copy of the lists.

This button allows the user to exit the PMS system and return to the login card of the driver.

The "MIP" card

The MIP card, for the computer implementation, has been divided into two different parts. The first has all the general information for the piece of equipment found on its existing paper counterpart. The second has the listing of all the MRC jobs along with a brief description, man/hour requirements, and other scheduling information.

MIP 1st page


MIP card

SYS. MIP CNTRL NO
2340/001-49

Main Propulsion Gas Turbines LM2500

SHIP SYSTEM SUBSYSTEM OR EQUIPMENT
Main Propulsion Gas Turbines LM2500
2340

REFERENCE PUBLICATIONS April 1989
S9234-AD-MMO-010/070

CONFIGURATION

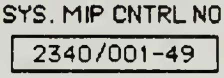
Incorporates SSDI System(s):
2345, Gas Turbine Lube Oil Storage and Conditioning Assembly
2341, Main Propulsion Gas Turbines LM2500
2342, Gas Turbine Base Enclosure
2343, Gas Turbine Free Standing Electronics
2344, Gas Turbine Water Wash Module Conditioning Assembly (See NOTE)
2346, Gas Turbine Bleed Air System
2347, Gas Turbine Starting Air System
2513, Combustion Air Intakes
2992, Special Tools, Propulsion

Page 1 of 2

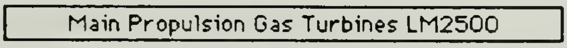
The fields

The fields found on this card are the same found on a actual MIP card . The only ones that differ are described below.

- *The "SYS. MIP CNTRL NO" field:* In the right top corner of the MIP card is the MIP number of this card. This number is located at the right bottom corner of the present hard copies of MIP cards.


SYS. MIP CNTRL NO
2340/001-49

- *The "Unit name" field:* This field does not exist on the paper implementation of MIP cards, but was added here to give to the user a quick reference to which piece of equipment they are looking at.


Main Propulsion Gas Turbines LM2500

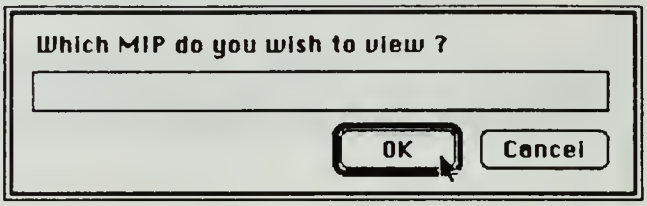
- *The "Page #" field:* The page field, gives the user a clue of how many cards this MIP is composed of and to which card he is currently on.

Page 1 of 2

The buttons

⇨ Gets the user page 2 of the MIP card.

This button invokes the search function. When pressed, the user is requested to enter an MIP number by the following dialog box.



Which MIP do you wish to view ?

The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

This button takes the user to the appropriate log card for this piece of equipment. In case that multiple pieces of the same equipment exist, the user is presented with the first occurance of this equipment in the log stack.

This button gets us to the appropriate piece of equipment in Argos's repair part section.

This button is a link to the Technical library. The current card is temporarily stored away and the user is presented with the Technical library stack.

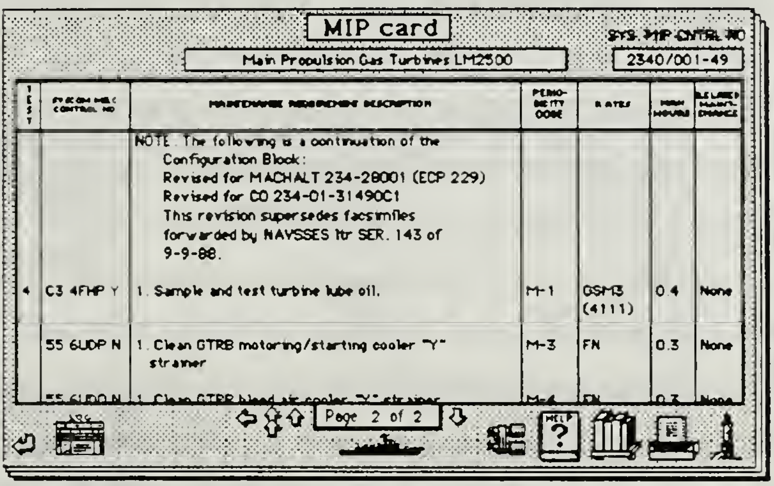
The print button prints the current MIP card. More information on this function is found under the "Auxilliary stacks'" description of this manual.

⏎ This button returns the user to the card they were on proir to being to presented this card.

This button allows the user to exit the PMS system and return to the login card of the driver.

MIP 2nd page

In this card we find the actual scheduling information for the particular piece of equipment.



MIP card

Main Propulsion Gas Turbines LM2500

SYS. MIP CNTRL NO 2340/001-49

TEST	SYSCOM MIP CONTROL NO	MAINTENANCE REQUIREMENT DESCRIPTION	PERIODICITY CODE	RATES	MAN HOURS	RELATED MAINT. CHANGE
		NOTE. The following is a continuation of the Configuration Block: Revised for MACHALT 234-28001 (ECP 229) Revised for CO 234-01-31490C1 This revision supersedes facsimiles forwarded by NAVSSES ltr SER. 143 of 9-9-88.				
4	C3 4FHP Y	1. Sample and test turbine lube oil.	M-1	GSM3 (4111)	0.4	None
	55 6UDP N	1. Clean GTRB motoring/starting cooler "Y" strainer	M-3	FN	0.3	None

Page 2 of 2

The fields

The standard MIP fields are found on this card. The ones that differ are described below.

- *The "SYS. MIP CNTRL NO" field:* Same as above.
- *The "Unit name" field:* Same as above.
- *The "Page #" field:* Same as above.

The buttons

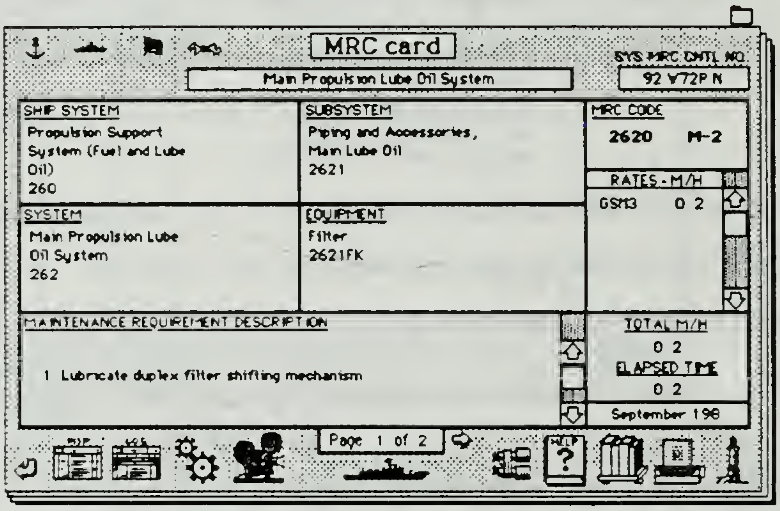
The buttons on this card are the same described on the MIP 1st page above.

The MRC card

The MRC card, for the computer implementation, has been divided into four different parts. The first has all the general information for the piece of equipment, as found on its existing paper counterpart. The second has the procedure part. The third, if it exists, has diagrams. The fourth contains tables. Multiple cards of each can exist for each MRC card.

MRC 1st page

This page contains all the general information on the piece of equipment, along with information required for automatic job scheduling and for the linking to the different places in the other databases of the system.



MRC card

Main Propulsion Lube Oil System

SYS. MRC CNTL NO.
92 W72P N

SHIP SYSTEM
Propulsion Support
System (Fuel and Lube
Oil)
260

SUBSYSTEM
Piping and Accessories,
Main Lube Oil
2621

MRC CODE
2620 M-2

RATES - M/H
GSM3 0 2

SYSTEM
Main Propulsion Lube
Oil System
262

EQUIPMENT
Filter
2621FK

MAINTENANCE REQUIREMENT DESCRIPTION
1 Lubricate duplex filter shifting mechanism

TOTAL M/H
0 2

ELAPSED TIME
0 2

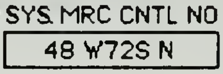
September 198

Page 1 of 2

The fields

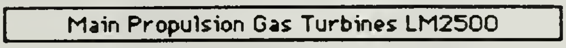
The fields found on this card are the same as found on the actual MIP card. Only the ones that differ are described below.

- *The "SYS. MRC CNTRL NO" field:* In the right top corner of the MRC card is the MRC number of this card.



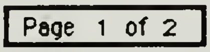
SYS. MRC CNTL NO
48 W72S N

- *The "Unit name" field:* This field does not exist on the paper implementation of MRC cards, but was added to give the user a quick reference as to which piece of equipment he is looking at.



Main Propulsion Gas Turbines LM2500

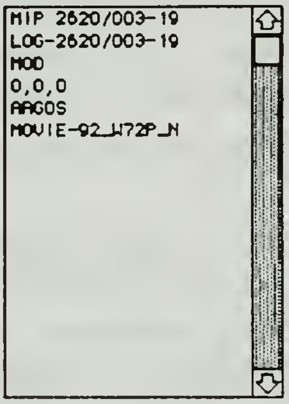
- *The "Page #" field:* The page field, provides an idea of how many cards this MRC is composed of, and which card is currently showing.



Page 1 of 2

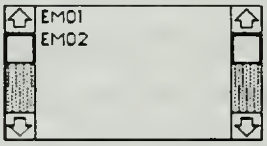
- *The "Data, work centers involved and relative maintenance "fields:* These are three invisible fields. They are show to the user when the 📁 icon at the right top corner of the MRC card is pressed. The icon is a toggle, changes to 📂 and needs to be pressed again in order to hide it. These fields are used for scheduling the jobs and for passing information to the various functions of the system. The most important of all is the data field.

The "Data" field: This field contains six lines(more lines can be added in the future if need arises). The first line is the link to the corresponding MIP card. The information is filled in automatically at the time the MRC card is created. It is very important that it has the format of "MIP", space and "MIP#". The second line contains information about the corresponding log card for this piece of equipment. Its format is "LOG-" and the corresponding MIP number. The third line is the MOD function result required for scheduling some jobs. This number determines whether a job should be scheduled during a particular quarter for jobs with periodicity longer than "quarterly" or on an even or odd month for "two month" jobs. It allows the system to make the decision, without keeping information stored about jobs scheduled during previous quarters. It also increases the reliability of the system and allows the SuperUser to determine which quarter a job will be scheduled, thus avoiding scheduling during the same quarter, the jobs with the same periodicity code. The fourth line contains three numbers separated by ",". The first of these numbers determines, how many cards of "MRC 2nd page" format this card is composed of in case the procedure section is too long to fit in one. The second number determines the number of graphics pages and the third represents the number of tables that belong to this card. The fifth line is the connection to the Argos stack. The sixth is the link to the corresponding picture in the "Movies" stack, if one exists. The format for this line is: "MOVIE-" and the "MRC #" with no spaces, but with underscores inserted instead of them.



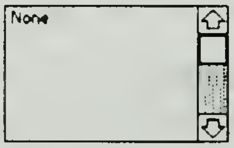
MIP 2620/003-19
LOG-2620/003-19
MOD
0,0,0
ARGOS
MOVIE-92_W72P_N

The "Work centers involved" field: This field contains the names of the works centers responsible for this particular MRC job. Information is either entered by hand, one work center per line, or automatically through the "Assign responsibility" function of the "PMS Toolbox".



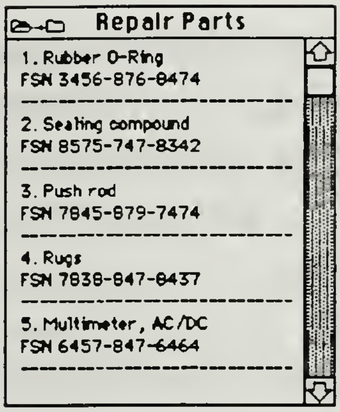
EM01
EM02

The *"Relative maintenance" field:* In this field we find either the word "None" or a list of MRC jobs that represent the relative maintenance for this job.



None

- *The "Repair parts" field:* This field is an invisible field activated with the ⚙ button. It contains the repair parts that might be needed to complete the particular job, along with their corresponding FSN. The 📂→📁 button closes the field.


Repair Parts

1. Rubber O-Ring
FSN 3456-876-8474

2. Sealing compound
FSN 8575-747-8342

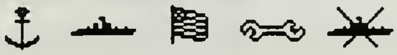
3. Push rod
FSN 7845-879-7474

4. Rugs
FSN 7838-847-8437

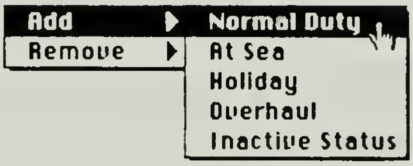
5. Multimeter, AC/DC
FSN 6457-847-6464

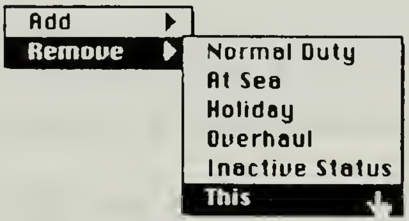
The buttons

The five buttons presented below appear on the upper left hand corner of the card and specify during which ship status the job may be performed. If a job may not be performed when the ship's status is one of these categories, we have to remove the appropriate icon.



When one of these buttons is pressed, a pop up menu appears and the user can select either to add a status or remove one. In the "remove" selection, the option "This" is included refers to the status button from which was invoked. These two menus are shown below.





The information from these buttons is used during the automatic job scheduling, to ensure that a job that has to be done when the ship is under way, i.e., "max speed achievement", is not scheduled on a day that the ship is inport. The five buttons represent the five "Ship Status" categories used throughout the PMS system. These are, from left to right, "In Port", "Under Way", "Major Holiday", "Major Overhaul","Inactive Maintenance".

⇨ Gets the user to page 2 of the MRC card.

MIP

This button gets the user to the corresponding MIP card this MIP belongs to.

LOG

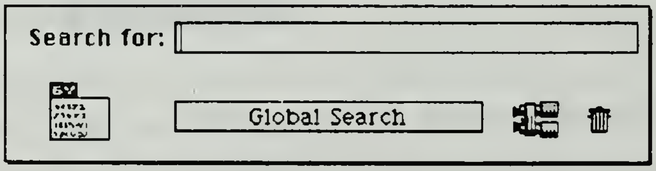
This button shows the user the appropriate log card for this piece of equipment for which the current MRC card refers to. In case that multiple pieces of the same equipment exist, the user is presented with the first occurrence of this equipment in the log stack.

This button presents to the user the field that contains the repair parts required for the particular job.

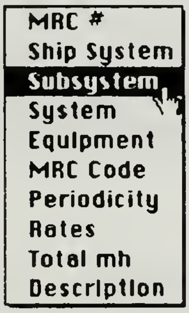
By pressing this button, the user can go to the appropriate picture that is relevant to the job, in the "Movies" stack.

This button gets us to the appropriate piece of equipment in Argos's repair part section.

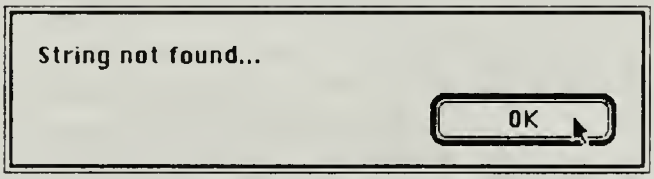
This button invokes the search function of the MRC card. When pressed, the dialog box, shown below, is presented to the user.



This dialog box is different from the ones presented already, because it has fields and buttons of its own. The "*Search for*" field is provided so that the user can type in the string to search for. The field on the bottom of the dialog is used to specify the kind of search the user wants to perform. The default setting is "Global Search". To change this and restrict the search to one specific field, press the button in order to invoke the pop up menu. This pop up menu, shown below, has the names of all fields to search on. When the user presses the mouse button to invoke the menu, the pointer is set on the "Subsystem" selection, which is the most likely, the one the user is going to select most of the times.



After specifying both the search key and the field to search on, press the button or the return key, and the search begins. If the key is not found, the user is informed by the following dialog box.


String not found...

In case the user changes his mind about doing a search he presses the button to remove the search dialog box.

The "Help" button invokes the on-line context sensitive help function of the system. More information about this function can be found on the description of the "Help" stack of this manual.

This button is a link to the Technical library. The current card is temporarily stored away and the user is presented with the Technical library stack.

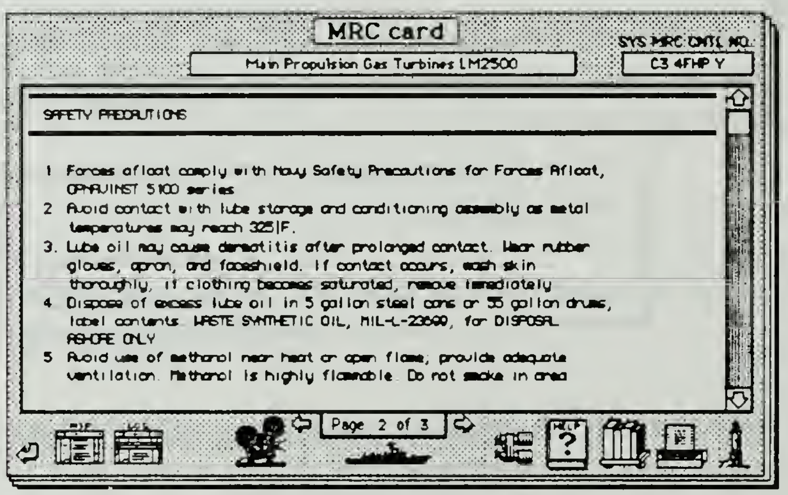
The print button prints the current MRC card. More information on this function is found under the "Auxiliary stacks'" description of this manual.

This button returns the user to the card they were on when they requested from the system to present this card.

This button allows the user to exit the PMS system and return to the login card of the driver.

MRC 2nd page

This page contains the procedure for performance of the particular job.



Main Propulsion Gas Turbines LM2500

C3 4FHP Y

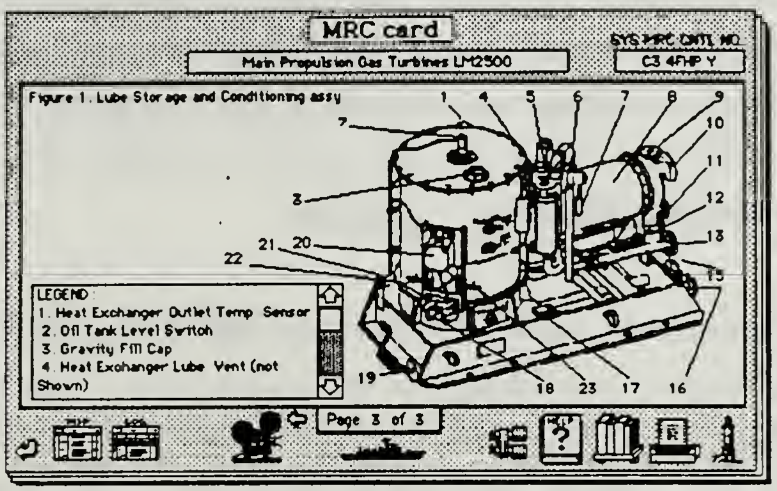
SAFETY PRECAUTIONS

1 Forces afloat comply with Navy Safety Precautions for Forces Afloat, OPNAVINST 5100 series
2 Avoid contact with lube storage and conditioning assembly as metal temperatures may reach 325|F.
3. Lube oil may cause dermatitis after prolonged contact. Wear rubber gloves, apron, and faceshield. If contact occurs, wash skin thoroughly; if clothing becomes saturated, remove immediately
4 Dispose of excess lube oil in 5 gallon steel cans or 55 gallon drums, label contents: WASTE SYNTHETIC OIL, MIL-L-23699, for DISPOSAL ASHORE ONLY
5 Avoid use of methanol near heat or open flame; provide adequate ventilation. Methanol is highly flammable. Do not smoke in area

In a few of the MRC cards, the procedure is so long that it had to be split into two second pages. Corresponding fields and buttons have the same function as on the "MRC 1st page" described in detail above.

MRC 3rd page

This page can consist of multiple pages also. The graphics, if any exist that, correspond to the particular MRC card. Corresponding fields and buttons have the same functions as on the "MRC 1st page" described in detail above.



Main Propulsion Gas Turbines LM2500

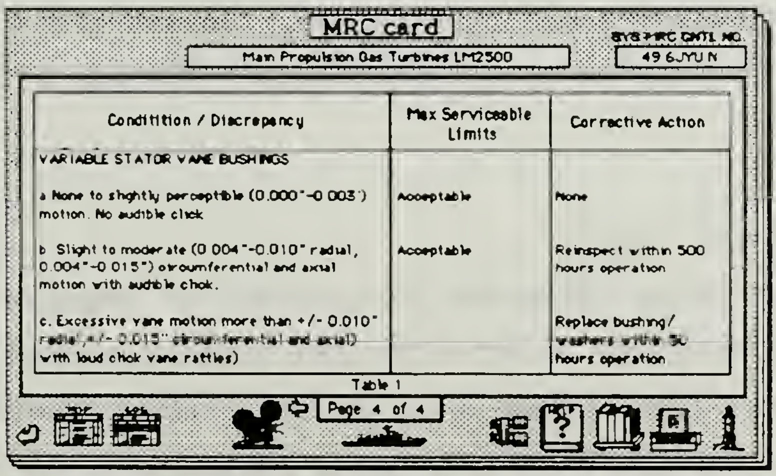
C3 4FHP Y

Figure 1. Lube Storage and Conditioning assy

LEGEND:
1. Heat Exchanger Outlet Temp Sensor
2. Oil Tank Level Switch
3. Gravity Fill Cap
4. Heat Exchanger Lube Vent (not Shown)

MRC 4th page

This page is a free format page where information not following any specific format can be placed. Tables or diagrams and graphics that do not require a legend can be stored on this page. Corresponding fields and buttons have the same function as on the "MRC 1st page" described in detail above.



Main Propulsion Gas Turbines LM2500

49 6JYU N

Condititon / Discrepancy	Max Serviceable Limits	Corrective Action
VARIABLE STATOR VANE BUSHINGS		
a None to slightly perceptible (0.000"-0.003") motion. No audible click	Acceptable	None
b Slight to moderate (0.004"-0.010" radial, 0.004"-0.015") circumferential and axial motion with audible chok.	Acceptable	Reinspect within 500 hours operation
c. Excessive vane motion more than +/- 0.010" radial, +/- 0.015" circumferential and axial) with loud chok vane rattles)		Replace bushing/ washers within 50 hours operation

Table 1

The Auxiliary stacks

General description

The "Auxiliary" stacks is a collection of stacks that must be present in order for the PMS system to run properly. They are never seen or manipulated by the user. They are stacks whose size, context and function is exclusively controlled by the system. They serve the purposes of "printing" and "form and resource storing".

The "Resource Storing" stacks

This group consists of stacks that are used by the system for storing forms and resources. It consists of two stacks, for the present time. The "Home" stack and the "Templates" stack.

The "Home" stack

This stack is supplied with HyperCard when purchased. In order for the PMS module to work properly, the "Home" stack supplied with the system should be used instead, because modifications and additions have been made to it. If such a thing is not desirable, all the resources from the PMS "Home" stack should be transferred with a resource editing program to the "Home" stack desired to be used. Details on this are given in the installation section of this manual. This is the only stack from the ones supplied that doesn't have to be in the same folder (directory) with the rest of the PMS system.

The "Templates" stack

This stack, as its name proposes, is a collection of templates of the various cards found throughout the system. When a new card must be generated, for example to add a new entry to the log stack, a copy of the log card is retrieved from the templates stack, prepared by the user and then stored into the log database. The system's function all work in a similiar way when the need to create a card of a new form arises.

If in the future a need arises to add some more functionality to the PMS system, a card of the new form should be added to this stack too.

The "General Label Storage" stack

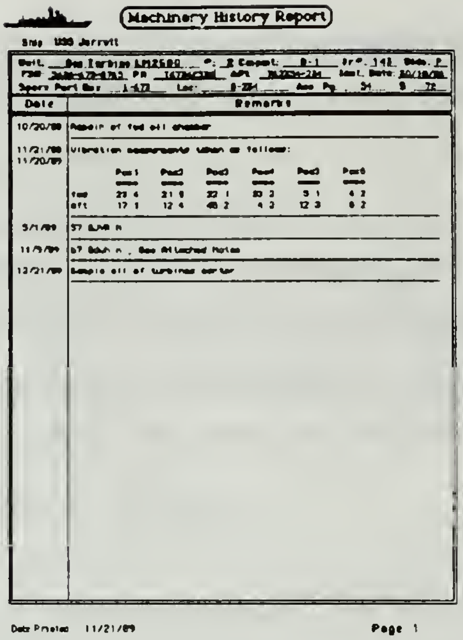
This stack is the holding tank for the bar coded labels, in case the user does not want to store them in a different stack of his own. It would be advisable to create a new stack of this kind for each situation. For example, when the ship goes for a major overhaul, a stack of this type should be generated in order to hold information about the material that went off from the ship, during this period. Another appropriate reason to create a new stack would be to keep a separate one for each division or work center. The choice of organizing the label storage has been left to the user, according to the particular needs of the situation.

The "Printing" stacks

The "printing" stacks were developed in order to overcome the limitation HyperCard has on printing reports in a user controllable formatted way. The function of each of them is described below.

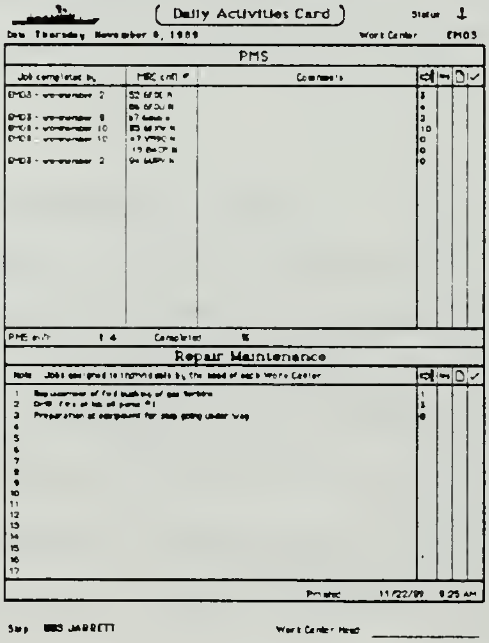
The "Log_report" stack

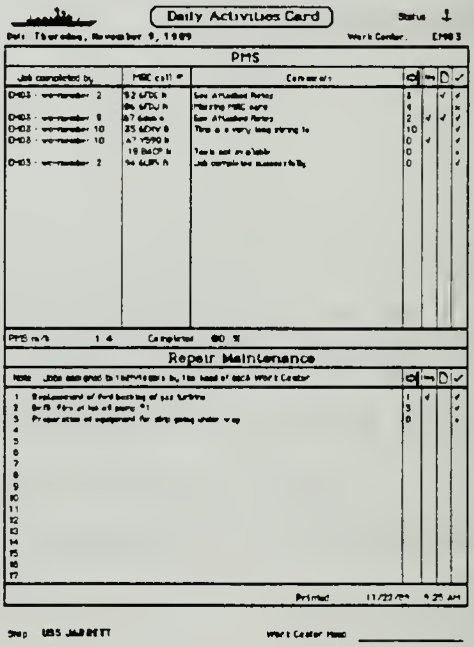
This stack, is used to print the log card of a piece of equipment. The system calculates the number of pages that are going to be needed to print the entire log card, and generates them accordingly. This way, the user is presented in hard copy with the same view seen on the computer's screen. A sample copy of a printed log card is presented below.


Machinery History Report
Date
Remarks
Page 1

The "DA_report" stack

This stack, comes to action when the user either requests a local print from a "Daily Activity" card or when he requests a hard copy remotely from the PMS_driver. If this function is requested from the PMS-driver level, the user has the option of producing either a DA card for one work center, or all the DA cards for every work center on the ship. This is the function that the SuperUser will use at the beginning of each day to produce the DA cards that the division officers will hand out to the work centers' supervisors, along with their specific instructions for the day.


Daily Activities Card
PMS
Repair Maintenance


Daily Activities Card
PMS
Repair Maintenance

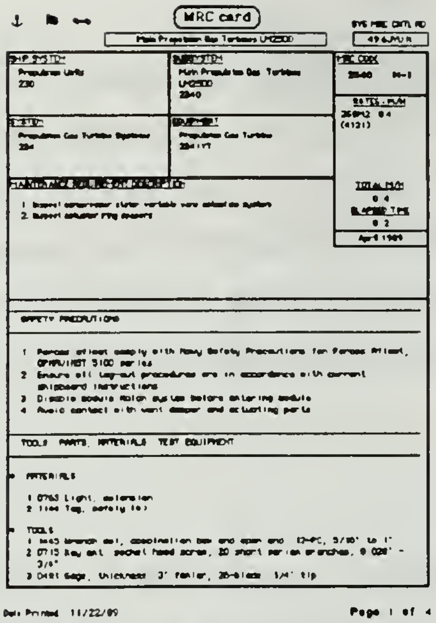
Apart for providing a hard copy with all the jobs to be done, it serves also as a printed report for future reference on such things as; what jobs have been done, whether they are transferred to the log stack, how many days their completion has been delayed, the amount of PMS m/h for the day, the ship's status and notes are associated with them. The form, generated at the beginning of the day, is shown on the left. The one that serves for reference after the day is over is presented on the right.

The "Generic_fld_print" stack

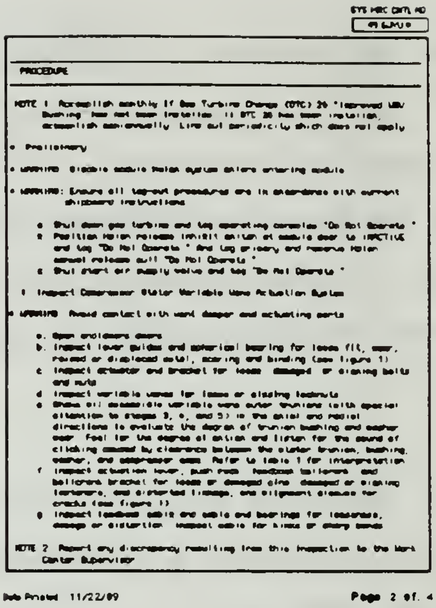
This stack is used to print various fields that do not have any specific format all around the PMS system. The reason for the existence of this stack, is because HyperCard's "Print Report" function has the capability to only print background fields in a acceptable for the PMS system way. The format they are printed in is not always suitable for our purposes. It is mainly used to print the "PMS Toolbox" fields for reference by the SuperUser.

The "MRC_print" stack

A complete PMS system must have the capability to reproduce the PMS cards on the fly. In cases of long deployments or when someone needs a copy of an MRC card to take along ,for reference or spot check purposes, MRC cards should be readily available and reproducible on demand. To provide this capability, this stack was introduced. The main reason is that every MRC card is unique and the format they come is not always consistent. To be precise, some cards have diagrams, some have tables, and some have both. Their length is not consistent either. By use of this stack, all the above inconsistencies are resolved by the system. Preserving the "information hiding" principle, the print function looks identical to the user through out the system. A typical MRC card is presented below. By varying the amount of pages, for each card, the system overcomes the problem that all current on-line PMS databases have. Which is to hold and reproduce only the text portion of MRC cards, without giving the capability to the user to get a "real" copy of the instructions of the job he has to perform.

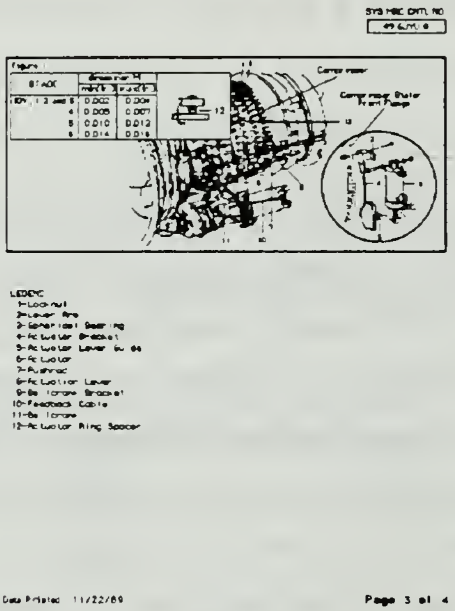


Date Printed 11/22/89 Page 1 of 4

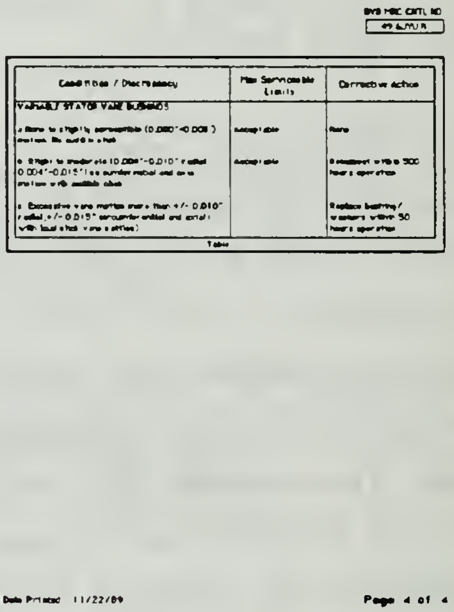


Date Printed 11/22/89 Page 2 of 4

If the MRC card has graphs and tables, they are added too.



Date Printed 11/22/89 Page 3 of 4

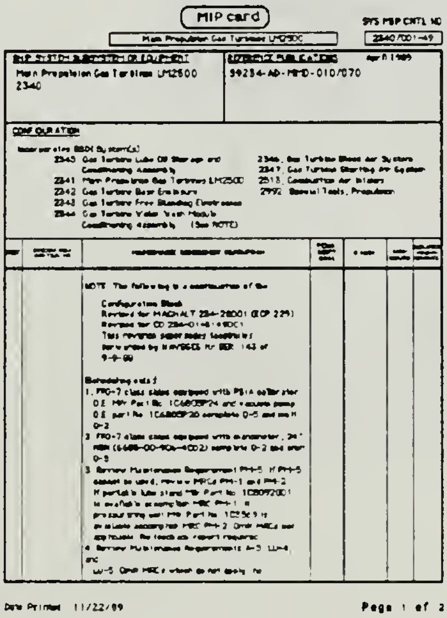


Date Printed 11/22/89 Page 4 of 4

This function is provided to all hands and no password restriction is applied, so as to encourage that personnel print cards, take them along and review them on their own pace. No "print all MRC cards" function has been included, to restrict the printing of all the near two thousand MRC cards currently in the PMS database. If such a function is needed in the future, the modularity of the system provides the capability to globally send a print message to the whole PMS database to implement such a job.

The "MIP_printed" stack

This stack is used to when a hard copy of an MIP card is required. It can also be invoked from the MIP card itself or the PMS_driver. It is of use mainly to the division officer or the PMS coordinator. This function is also capable of producing the amount of pages required for each MIP card. The format MIP cards are presented in is identical to the one used in the Navy. An example of a printed MIP card is presented below. This card consists of eight printed pages but was cut down to two for demo purposes.


MIP card

SYS MIP CNTL NO

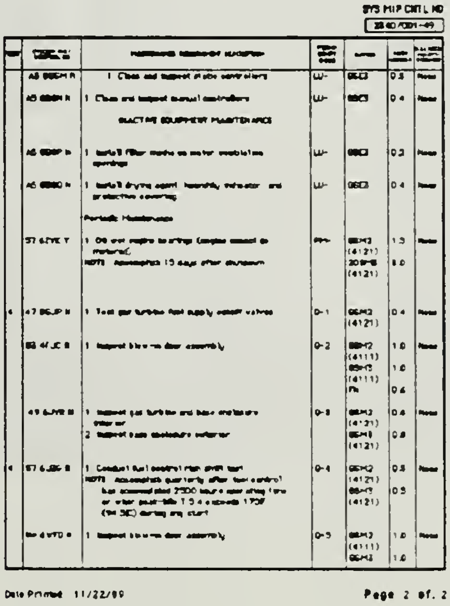
2340/001-49

Main Propulsion Gas Turbine LM2500

2340

Date Printed 11/22/89

Page 1 of 2


SYS MIP CNTL NO

2340/001-49

Date Printed 11/22/89

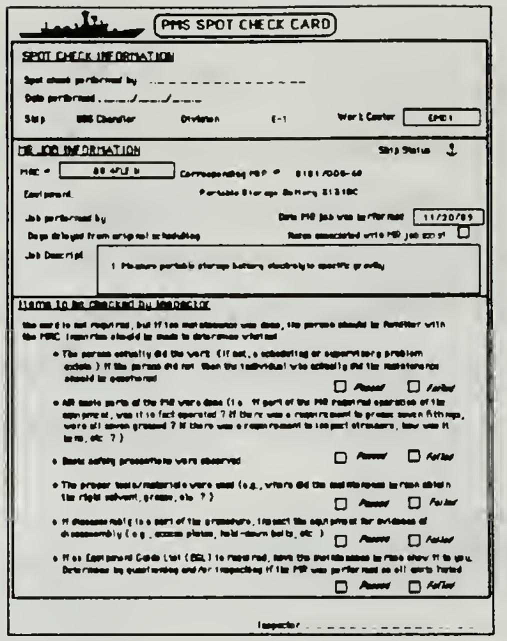
Page 2 of 2

The "Compl_report" stack

This stack is used to print the "% completed" graphs, as described in the report section of this manual.

The "Spot Check" stack

This stack is used to print the "spot check" form when one is requested. An example of the spot check form as printed on the printer is shown below. Along with the spot check card, the selected MRC card is printed too.



PMS SPOT CHECK CARD

SPOT CHECK INFORMATION

Spot check performed by ________________

Date performed ___/___/___

Ship USS Chandler Division E-1 Work Center EM01

MR JOB INFORMATION Ship Status ⚓

MRC # 80 4FLE N Corresponding MIP # 8181/006-60

Equipment Portable Storage Battery 313100

Job performed by Date MR job was performed 11/20/89

Days delayed from original scheduling Rates associated with MR job card ☐

Job Descript 1. Measure portable storage battery electrolyte specific gravity

Items to be checked by inspector

The card is not required, but if the maintenance was done, the person should be familiar with the MRC. Inquiries should be made to determine whether:

- The person actually did the work. (If not, a scheduling or supervisory problem exists.) If the person did not, then the individual who actually did the maintenance should be questioned. ☐ Passed ☐ Failed
- All main parts of the MR were done (i.e., if part of the MR required operation of the equipment, was it in fact operated? If there was a requirement to grease seven fittings, were all seven greased? If there was a requirement to inspect strainers, how was it done, etc.?) ☐ Passed ☐ Failed
- Basic safety precautions were observed ☐ Passed ☐ Failed
- The proper tools/materials were used (e.g., where did the maintenance person obtain the right solvent, grease, etc.?) ☐ Passed ☐ Failed
- If disassembly is a part of the procedure, inspect the equipment for evidence of disassembly (e.g., access plates, hold-down bolts, etc.) ☐ Passed ☐ Failed
- If an Equipment Guide List (EGL) is required, have the maintenance person show it to you. Determine by questioning and/or inspecting if the MR was performed on all units listed. ☐ Passed ☐ Failed

Inspector ________________

APPENDIX B

```
PMS_Module_Scripts        Thu Dec  7 15:07:06 1989        1

SCRIPTS FOR STACK: PMS_driver
_______________________________________________

****** STACK SCRIPT *****
on closeStack
  if the freesize of this stack > 0.15 * the size of this stack then
    doMenu "Compact Stack"
  end if
end closeStack

***** BACKGROUND #2: pms_driver_menus *****
function fieldsort thefield,sort_on_item
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield - 1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines in thefield
      if item sort_on_item of line current of thefield < item sort_on_item of line least
linenum of thefield then
        put current into leastlinenum
      end if
    end repeat
    if leastlinenum > primary then
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsort
 function CLICKLINE
  return trunc(((scroll of the target)
  + (item 2 of the clickloc) - (item 2 of the rect of the target
  )) div the textheight of the target) + 1
end CLICKLINE

on NewQuarter
  push card
  show button "Preparing New Calendar......."
  lock screen
  go to stack Ship_data
  put card fld crd_fld_Quart_aft_ovhl + 1 into card fld crd_fld_Quart_aft_ovhl
  put card fld crd_fld_Quart_aft_ovhl into Q_ovhl
  put 60*60*24 into OneDay                         --seconds in a day
  go to last card of stack "planning"
  put getStartDate() into start                    --what date to begin on
  repeat with cnt = 1 to 3                         --months in a quarter
    CreateNewMonth
    put start into it                              --a copy
    add OneDay*8 to it                             --skip first week
    convert it to long date
    put word 1 of item 2 of it && item 3 of it`
    into the_name
    put the_name into field "bg_fld_month_year"
    set the name of this card to the_name
    put Q_ovhl into fld bg_fld_qrter_after_ovhl
    Repeat with index = 1 to 37
      put start into it
      put it into change
      convert it to long date
      put (word 2 of item 2 of it) into field index
      convert it to date
      put "bg_fld_date_full_" & index into temp
      put it into fld temp
      add OneDay to start
    end repeat
    repeat with i=1 to 6
      if field i>1 then
        set the visible of card field i to true
      end if
    end repeat
    repeat with i=29 to 37
      if field i<i-10 then
        set the visible of card field (i-22) to true
      end if
    end repeat
    add OneDay*8 to start
    put startOfMonth(start) into start
  end repeat
  push card
  go to card PMS_driver_crd1 of stack PMS_driver
  send mouseUp to button "Preparing New Calendar......."
  pop card
  unlock screen
  Set_activities
end NewQuarter

on Set_activities
  go prev
  go prev
  answer "Set the Ship's activities for this Quarter" with "Cancel" or
  "OK"
  if it is "Cancel" then
    repeat 3 times
      doMenu Delete Card
    end repeat
    pop card
  end if
end Set_activities

function getStartDate
  put word 1 of field "bg_fld_month_year" && "1," && `
  word 2 of field "bg_fld_month_year" into it
  convert it to seconds
  add (60*60*24*32) to it
  put startOfMonth(it) into it     --so we can compare
  put the date into thisMonth
  put startOfMonth(thisMonth) into thisMonth   --both at week starts in seconds
  if it >= thisMonth then return it   --calendar already beyond today
  convert it to seconds
  return it
end getStartDate

function startOfMonth aDate      -- the seconds of Monday before the 1st
  put 60*60*24 into OneDay       --seconds in a day
  put aDate into it
  convert it to long date
  put 1 into word 2 of item 2 of it      --first of month
  convert it to seconds
  convert it to long date
  repeat while item 1 of it <> "Sunday"
    convert it to seconds
    subtract OneDay from it    --look for previous Sunday
    convert it to long date
  end repeat
  convert it to seconds
  return it
end startOfMonth

on CreateNewMonth
  push card
  go to card "monthly_calendar_model card" of stack templates
  doMenu "Copy Card"
  pop card
  doMenu "Paste Card"
end CreateNewMonth

on do_report
  global WORK_CENTERS_TO_REPORT
  global START_MONTH
  global TEMP_YEAR
  global PRINT_ME
  global GO_Q
  put 1 into STORE
  repeat with WORK_C = 1 to the number of items of WORK_CENTERS_TO_REPORT
    put item WORK_C of WORK_CENTERS_TO_REPORT into TEMPWC
    put empty into DATA_M1
    put empty into DATA_M2
    put empty into DATA_M3
    -- go to desired stack
    go to stack GO_Q
    -- look for the first card of the desired wc
    go to card 1 of bg DA_model
    find TEMPWC in fld bg_fld_WC_DA
    -- get desired info for first card of month 1
    put char 1 of item 1 of fld bg_fld_date & ","
    & fld letter_sh_status & "," & fld completed & return
    after DATA_M1
    go next
    -- get desired info for the rest of cards of month 1
    repeat while word 2 of item 2 of fld bg_fld_date is not 1
      put char 1 of item 1 of fld bg_fld_date & ","
      & fld letter_sh_status & "," & fld completed & return
      after DATA_M1
      go next
    end repeat
    -- get desired info for first card of month 2
    put char 1 of item 1 of fld bg_fld_date & ","
    & fld letter_sh_status & "," & fld completed & return
    after DATA_M2
    go next
    -- get desired info for the rest of cards of month 2
    repeat while word 2 of item 2 of fld bg_fld_date is not 1
      put char 1 of item 1 of fld bg_fld_date & ","
      & fld letter_sh_status & "," & fld completed & return
      after DATA_M2
      go next
    end repeat
    push card
    go last
    put the short name of this card into TEMPNAME
    pop card
    -- get desired info for first card of month 3
    put char 1 of item 1 of fld bg_fld_date & ","
    & fld letter_sh_status & "," & fld completed & return
```

```
    go next
    -- get desired info for the rest of cards of month 3
    repeat while word 2 of item 2 of fld bg_fld_date is not 1 or
      the short name of this card = TEMPNAME
      put char 1 of item 1 of fld bg_fld_date & ","
      & fld letter_sh_status & "," & fld completed & return
      after DATA_M3
      go next
    end repeat
    -- go to pick a copy of graph paper...
    go to card graph_model_pms_compl of stack templates
    -- get it
    doMenu "Copy Card"
    -- go to print stack
    go to last card of stack compl_report
    -- make three pieces of graph paper
    repeat for 3
      doMenu "Paste Card"
    end repeat
    -- go to the first card
    go prev
    go prev
    -- fill in info
    put DATA_M1 into fld collect_info
    put TEMPWC into fld WC_rep
    put START_MONTH && "/" && TEMP_YEAR into fld month_rep
    go next
    put DATA_M2 into fld collect_info
    put TEMPWC into fld WC_rep
    put (START_MONTH + 1) && "/" && TEMP_YEAR into fld month_rep
    go next
    put DATA_M3 into fld collect_info
    put TEMPWC into fld WC_rep
    put (START_MONTH + 2) && "/" && TEMP_YEAR into fld month_rep
    go next
    repeat with i = 1 to 3
      go to card STORE + i
      send mouseUp to bg button make_me
    end repeat
    put STORE + 3 into STORE
  end repeat
  -- go to first card
  go first
  -- delete reminder from previous report...
  doMenu "Delete Card"
  -- check if output is printer
  if PRINT_ME then
    -- print all cards
    hide bg button print_only_me
    hide bg button lighthouse
    hide bg button go_next
    hide bg button go_prev
    hide bg button go_to_driver
    hide bg button "Performance Report"
    open printing
    print all cards
    close printing
    show bg button print_only_me
    show bg button lighthouse
    show bg button go_next
    show bg button go_prev
    show bg button go_to_driver
    go to card 2 of stack pms driver
    -- restore screen
    unlock screen
  end if
  put empty into WORK_CENTERS_TO_REPORT
  put empty into START_MONTH
  put empty into TEMP_YEAR
  put empty into PRINT_MF
  put empty into GO_Q
  -- play tune
  play "MOZART"
end do_report

on sp_all_wc
  put 1 into j
  put 1 into k
  put card field divlist1 into divlisttemp
  put the number of lines of card field divlist1 into divlisttemp1
  lock screen
  repeat for divlisttemp1
    put line j of divlisttemp into wcvar
    put wcvar into msg box
    if wcvar is in "EM01,EM06" then
      put "E-1" into divvar
    else if wcvar is in "EM02,EM03,EM04" then
      put "E-2" into divvar
    else if wcvar is in "EM05,ER09" then
      put "E-3" into divvar
    else if wcvar is in "CS01,CS02" then
      put "CS-1" into divvar
    else if wcvar is in "CS03" then
      put "CS-2" into divvar
    else if wcvar is in "CS04,CS05,CS06" then
      put "CS-3" into divvar
    else if wcvar is in "CS07,CS08,CS09,CS10" then
      put "CS-4" into divvar
    else if wcvar is in "OC01" then
      put "SC-1" into divvar
    else if wcvar is in "OC02" then
      put "SC-2" into divvar
    else if wcvar is in "OC03" then
      put "SC-3" into divvar
    else if wcvar is in "SS02" then
      put "S-1" into divvar
    else if wcvar is in "SS06" then
      put "S-2" into divvar
    else if wcvar is in "SS03,SS05,SS07" then
      put "S-3" into divvar
    else if wcvar is in "SS01,SS04" then
      put "S-4" into divvar
    end if
    put the abbrev date into datevar
    put word 1 of item 2 of datevar into dvar
    put item 3 of datevar into ytemp
    put char 4 to 5 of ytemp into ytemp
    put item 3 of datevar into yearvar
    if dvar is in "Jan,Feb,Mar" then
      put "DA Jan-Mar" & yearvar into atackvar
    else if dvar is in "Apr,May,Jun" then
      put "DA Apr-Jun" & yearvar into stackvar
    else  if dvar is in "Jul,Aug,Sep" then
      put "DA Jul-Sep" & yearvar into stackvar
    else  if dvar is in "Oct,Nov,Dec" then
      put "DA Oct-Dec" & yearvar into stackvar
```

```
    end if
    put char 1 to 3 of word 2 of stackvar into startdate
    put the short date into dattemp
    convert dattemp to seconds
    put dattemp - (60 * 60 * 24* 14) into starttemp
    put dattemp - starttemp into diftemp
    convert starttemp to long date
    divide diftemp by (60 * 60 * 24)
    put random (diftemp) into rtemp
    put random (7) into chkvar
    if k = 1 then
      go to card 22 of stack stackvar
      put k + 1 into k
    end if
    find wcvar in background field id 92
    find starttemp in background field id 114
    repeat for rtemp
      go next
    end repeat
    put false into resultvar
    repeat for rtemp + 1
      put chkvar into test
      repeat for chkvar
        put line test of bg fld id 165 into varchk
        if varchk = "\326" then
          put true into resultvar
          put line test of bg fld bg_fld_MRC_no into MRCvar
          put line test of bg fld bg_fld_job_compl_by into Jobvar
          put line test of bg fld bg_fld_PMS_days_fwd into delvar
          put line test of bg fld bg_fld_note_activator into notevar
          put bg fld bg_fld_date into dayvar
          convert dayvar to short date
          put bg fld letter_sh_status into letter_var
          exit repeat
        end if
        put (test - 1) into test
      end repeat
      if resultvar = true then
        exit repeat
      else
        go previous
      end if
    end repeat
    if resultvar = false then
      -- answer "NO JOBS COMPLETED for W/C " &  wcvar with "OK"
      put j + 1 into j
      next repeat
    end if
    push card
    go to card MRCvar of stack PMS
    put word 2 of line 1 of bg fld data into MIPvar
    put bg fld mr_description_field into mrvar
    put bg fld equipment_field into equipvartemp
    put the number of words of equipvartemp into equipno
    repeat with i = 1 to equipno
      put word i of equipvartemp & " " after equipvar
    end repeat
    go to stack templates
    go to card "spot_ck_1"
    doMenu "COPY CARD"
    go to stack spot_check
    doMenu "LAST"
    doMenu "PASTE CARD"
    put wcvar into bg fld work_center
    put divvar into bg fld division
    put MRCvar into bg fld "MRC_#"
    put MIPvar into bg fld "MIP_#"
    put jobvar into bg fld performed_by
    put dayvar into bg fld date
    if delvar is empty then
      put "" into bg fld delayed
    else
      put delvar into bg fld delayed
    end if
    if mrvar is empty then
      put "" into bg fld job_description
    else
      put mrvar into bg fld job_description
    end if
    if equipvar is empty then
      put "" into bg fld equipment
    else
      put equipvar into bg fld equipment
    end if
    if notevar <> empty then
      set the hilite of bg btn notes to true
    end if
    if letter_var = "A" then
      show bg btn anchor
    else if letter_var = "B" then
      show bg btn at_sea
    else if letter_var = "C" then
      show bg btn flag
    else if letter_var = "D" then
      show bg btn overhaul
    else if letter_var = "E" then
      show bg btn inactive
    end if
    go to stack templates
    go to card "spot_ck_2"
    doMenu "COPY CARD"
    go to stack spot_check
    doMenu "LAST"
    doMenu "PASTE CARD"
    put j + 1 into j
    pop card
    put empty into equipvar
    put empty into wcvar
    put empty into divvar
    put empty into MRCvar
    put empty into MIPvar
    put empty into jobvar
    put empty into dayvar
    put empty into delvar
    put empty into mrvar
  end repeat
  go to card 1 of stack spot_check
  put the number of cards into delvar
  domenu "DELETE CARD"
  domenu "DELETE CARD"
  print all cards
  repeat for delvar - 2
    go to card 1
    domenu "DELETE CARD"
  end repeat
  go to card 2 of "PMS_driver"
```

```
end sp_all_wc

on sp_1_wc
  -- if "For one WC" was selected...
  --lock screen
  show card fld divlist
  show card fld divselect
  unlock screen
end sp_1_wc

 ***** BKGND #2, BUTTON #1: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #2, BUTTON #4: return *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #2, BUTTON #8: MRC Spot Check *****
on mouseup
  global wcvar,current_Q
  put false into mrcopy
  answer "DO YOU WANT A COPY OF THE MRC" with "YES" or "NO"
  if it = "YES" then
    put true into mrcopy
  end if
  if wcvar is in "EM01,EM06" then
    put "E-1" into divvar
  else if wcvar is in "EM02,EM03,EM04" then
    put "E-2" into divvar
  else if wcvar is in "EM05,EP09" then
    put "E-3" into divvar
  else if wcvar is in "CS01,CS02" then
    put "CS-1" into divvar
  else if wcvar is in "CS03" then
    put "CS-2" into divvar
  else if wcvar is in "CS04,CS05,CS06" then
    put "CS-3" into divvar
  else if wcvar is in "CS07,CS08,CS09,CS10" then
    put "CS-4" into divvar
  else if wcvar is in "OC01" then
    put "SC-1" into divvar
  else if wcvar is in "OC02" then
    put "SC-2" into divvar
  else if wcvar is in "OC03" then
    put "SC-3" into divvar
  else if wcvar is in "SS02" then
    put "S-1" into divvar
  else if wcvar is in "SS06" then
    put "S-2" into divvar
  else if wcvar is in "SS03,SS05,SS07" then
    put "S-3" into divvar
  else if wcvar is in "SS01,SS04" then
    put "S-4" into divvar
  end if
  put the abbrev date into datevar
  put word 1 of item 2 of datevar into dvar
  put item 3 of datevar into ytemp
  put char 4 to 5 of ytemp into ytemp
  --else if dvar is in "Apr,May,Jun" then
  --   put "DA Apr-Jun" & yearvar into stackvar
  -- else  if dvar is in "Jul,Aug,Sep" then
  --  put "DA Jul-Sep" & yearvar into stackvar
  -- else  if dvar is in "Oct,Nov,Dec" then
  --   put "DA Oct-Dec" & yearvar into stackvar
  -- end if
  put current_Q into stackvar
  put char 1 to 3 of word 2 of stackvar into startdate
  lock screen
  put the short date into dattemp
  convert dattemp to seconds
  put dattemp - (60 * 60 * 24* 14) into starttemp
  put dattemp - starttemp into diftemp
  convert starttemp to long date
  divide diftemp by (60 * 60 * 24)
  put random (diftemp) into rtemp
  put random (7) into chkvar
  go to card 22 of stack stackvar
  find wcvar in background field id 92
  find starttemp in background field id 114
  repeat for rtemp
    go next
  end repeat
  put false into resultvar
  repeat for rtemp + 1
    put chkvar into test
    repeat for chkvar
      put line test of background field id 165 into varchk
      if varchk = "\326" then
        put true into resultvar
        put line test of bg fld bg_fld_MRC_no into MRCvar
        put line test of bg fld bg_fld_job_compl_by into Jobvar
        put line test of bg fld bg_fld_PMS_days_fwd into delvar
        put line test of bg fld bg_fld_note_activator into notevar
        put bg fld bg_fld_date into dayvar
        convert dayvar to short date
        put bg fld letter_sh_status into letter_var
        exit repeat
      end if
      put (test - 1) into test
    end repeat
    if resultvar = true then
      exit repeat
    else
      go previous
    end if
  end repeat
  if resultvar = false then
    answer "NO JOBS COMPLETED" with "OK"
    go to card 2 of stack "PMS_driver"
    unlock screen
    exit mouseup
  end if
  go to card MRCvar of stack PMS
  put word 2 of line 1 of bg fld data into MIPvar
  put bg fld mr_description_field into mrvar
  put bg fld equipment_field into equipvartemp
  put the number of words of equipvartemp into equipno
  repeat with i = 1 to equipno
    put word i of equipvartemp & " " after equipvar
  end repeat
  go to stack templates
  go to card "spot_ck_1"
  doMenu "COPY CARD"
  go to stack spot_check
  doMenu "LAST"
  doMenu "PASTE CARD"
  put wcvar into bg fld work_center
  put divvar into bg fld division
  put MRCvar into bg fld "MRC_#"
  put MIPvar into bg fld "MIP_#"
  put jobvar into bg fld performed_by
  put dayvar into bg fld date
  put delvar into bg fld delayed
  put mrvar into bg fld job_description
  put equipvar into bg fld equipment
  if notevar <> empty then
    set the hilite of bg btn notes to true
  end if
  if letter_var = "A" then
    show bg btn anchor
  else if letter_var = "B" then
    show bg btn at_sea
  else if letter_var = "C" then
    show bg btn flag
  else if letter_var = "D" then
    show bg btn overhaul
  else if letter_var = "E" then
    show bg btn inactive
  end if
  go to stack templates
  go to card "spot_ck_2"
  doMenu "COPY CARD"
  go to stack spot_check
  doMenu "LAST"
  doMenu "PASTE CARD"
  go to card 1 of stack spot_check
  domenu "DELETE CARD"
  domenu "DELETE CARD"
  print all cards
  if mrcopy = true then
    go to card MRCvar of stack PMS
    send mouseup to background button print_mrc
  end if
  go to card 2 of stack "PMS_driver"
  unlock screen
end mouseup

***** CARD #1: PMS_driver_open *****
on opencard
  hide menubar
  global CHECK_DESIRE
  global DRIV_GENERAL_MENU
  global DRIV_ENGINEERING_MENU
  global DRIV_SUPPORT_MENU
  global DRIV_SHIP_CONTROL_MENU
  global DRIV_COMBAT_SYS_MENU
  global LOG_MFCTR_DATA_MENU
  global LOG_PUBS_MENU
  global LOG_DRWG_NUM_MENU
  global LOG_DATE_MENU
  global LOG_REMARK_MENU
  global DA_PMS_MENU
  global DA_REP_MAINT_MENU
  global CURRENT_Q
  global OPEN_STACKS
  -- supply access
  put TRUE into OPEN_STACKS
  -- ask for the user's password
  play "PASSWORD PLEASE"
  --prepare PMS_driver's menus
  -- prepare DRIV_GENERAL_MENU
  put "(Password,(Add new member,(Remove a member,(Change members access,(Change my pass
word" into DRIV_GENERAL_MENU
  put return & "(Schedule,(Next quarter"  after DRIV_GENERAL_MENU
  put return & "(Reports & messages,(% done diagram - 1 WC,(% done diageam - all WC,(Bar
 coded label,(Navy message,(Ext. rep req" after DRIV_GENERAL_MENU
  put return & "(Spot checks,(For entire ship,(For one WC" after DRIV_GENERAL_MENU
  put return & "(Go,(MRC card,(MIP card,(Log card"  after DRIV_GENERAL_MENU
  put return & "(Print,(Today's DA cards - all,(Today's DA card - 1 WC,(MIP card,(MRC ca
rd,(Log card" after DRIV_GENERAL_MENU
  put return & "(Toolbox,(Add more MIP cards,(Convert MIP-MRC file to stack,(Empty PMS d
atabase,(MIP-MRC cards in PMS database,(Delete a MIP set,(WC responsibility" after DRIV_
GENERAL_MENU
  put return & "(Equipment log,(Add entry,(Del entry"  after DRIV_GENERAL_MENU
  --prepare DRIV_ENGINEERING_MENU
  put "(E-1,(EM01,(EM06" into DRIV_ENGINEERING_MENU
  put return & "(E-2,(EM02,(EM03" after DRIV_ENGINEERING_MENU
  put return & "(E-3,(EM04,(EM05,(ER09" after DRIV_ENGINEERING_MENU
  --prepare DRIV_SUPPORT_MENU
  put "(S-1,(SS02" into DRIV_SUPPORT_MENU
  put return & "(S-2,(SS06" after DRIV_SUPPORT_MENU
  put return & "(S-3,(SS03,(SS05,(SS07" after DRIV_SUPPORT_MENU
  put return & "(S-4,(SS01,(SS04" after DRIV_SUPPORT_MENU
  --prepare DRIV_SHIP_CONTROL_MENU
  put "(SC-1,(OC01" into DRIV_SHIP_CONTROL_MENU
  put return & "(SC-2,(OC02" after DRIV_SHIP_CONTROL_MENU
  put return & "(SC-3,(OC03" after DRIV_SHIP_CONTROL_MENU
  --prepare DRIV_COMBAT_SYS_MENU
  put "(CS-1,(CS01,(CS02" into DRIV_COMBAT_SYS_MENU
  put return & "(CS-2,(CS03" after DRIV_COMBAT_SYS_MENU
  put return & "(CS-3,(CS04,(CS05,(CS06" after DRIV_COMBAT_SYS_MENU
  put return & "(CS-4,(CS07,(CS08,(CS09" after DRIV_COMBAT_SYS_MENU
  --prepare Log stack's menus
  -- prepare LOG_MFCTR_DATA_MENU
  put "(Dial mfctr" into LOG_MFCTR_DATA_MENU
  put return & "(Modify mfct data" after LOG_MFCTR_DATA_MENU
  -- prepare LOG_PUBS_MENU
  put "(Go to pub" into LOG_PUBS_MENU
  put return & "(Modify pubs" after LOG_PUBS_MENU
  -- prepare LOG_DRWG_NUM_MENU
  put "(Go to drawing" into LOG_DRWG_NUM_MENU
  put return & "(Modify drawings" after LOG_DRWG_NUM_MENU
  -- prepare LOG_DATE_MENU
  put "(Enter today's date" into LOG_DATE_MENU
  put return & "(Remove date" after LOG_DATE_MENU
  put return & "(Enter other date" after LOG_DATE_MENU
  -- prepare LOG_REMARK_MENU
  put "(Add data" into LOG_REMARK_MENU
  put return & "(Modify remarks" after LOG_REMARK_MENU
  --prepare DA cards' menus
  -- prepare DA_PMS_MENU
  put "(Check job,(Done,(Cancel check,(Cancel PMS job" into DA_PMS_MENU
  put return & "(Notes,(Fill,(See,(See all" after DA_PMS_MENU
  put return & "(Equipment log,(Go to log,(Add job to log" after DA_PMS_MENU
  put return & "(Go,(MRC card,(MIP card" after DA_PMS_MENU
  put return & "(Comment,Job completed succesfully,Missing MRC card,Tools not available,
```

```
ENU
  put return & "(Check repair job,(Done,(Cancel check,(Cancel repair job" after DA_REP_MA
INT_MENU
  -- prepare change password flag
  put 1234 into CHECK_DESIRE
  put empty into card fld abcd of card PMS_driver_crd1
  -- find current quarter
  put the date into TO_EXTR
  convert TO_EXTR to dateItems
  -- check what quarter we are on
  if item 2 of TO_EXTR = 1 or item 2 of TO_EXTR = 2 or item 2 of TO_EXTR = 3 then
    put "DA JAN-MAR" && item 1 of TO_EXTR into CURRENT_Q
  end if
  if item 2 of TO_EXTR = 4 or item 2 of TO_EXTR = 5 or item 2 of TO_EXTR = 6 then
    put "DA APR-JUN" && item 1 of TO_EXTR into CURRENT_Q
  end if
  if item 2 of TO_EXTR = 7 or item 2 of TO_EXTR = 8 or item 2 of TO_EXTR = 9 then
    put "DA JUL-SEP" && item 1 of TO_EXTR into CURRENT_Q
  end if
  if item 2 of TO_EXTR = 10 or item 2 of TO_EXTR = 11 or item 2 of TO_EXTR = 12 then
    put "DA OCT-DEC" && item 1 of TO_EXTR into CURRENT_Q
  end if
end opencard

on returnkey
  -- set the msg box out of sight
  set the loc of msg to 22,-500
  if msg box = empty then
    -- no password typed and therefore repeat password request
    send opencard to this card
  else
    -- if some password was typed
    put msg into PRESERVE_PWD
    put msg into card fld abcd of card PMS_driver_crd1
    put empty into msg
    hide msg
    set the loc of msg to 22,300
    -- inform the user
    -- if the first time password was given
    if PRESERVE_PWD is "me" then
      ask "Give me your last name please."
      put it into NAME_CHECK
      if NAME_CHECK is not empty then
        show button "Checking password database..."
        lock screen
        push card
        -- go to check whether a card is set up for this person
        go to stack "Password stack"
        find whole NAME_CHECK in fld bg_fld_rank_name
        -- if person not yet in the database or somebody with same name
        -- allready there
        if the result is not empty or fld dwp is not "me" then
          set the lockMessages to true
          pop card
          hide button "Checking password database..."
          show button "Access denied !!!"
          beep
          unlock screen
          set the lockMessages to false
          wait 4 seconds
          put the short id of this card into TEMP_ID
          pop card
          unlock screen
          set the lockMessages to false
          hide button "Checking password database..."
          ask "What is your new password ?"
          if it is not empty then
            put it into PWD1
            ask "Type it again please..."
            if it = PWD1 then
              show button "Setting password..."
              lock screen
              push card
              go to card id TEMP_ID of stack "Password stack"
              put PWD1 into fld dwp
              set the lockMessages to true
              pop card
              hide button "Setting password..."
              unlock screen
              set the lockMessages to false
              send opencard to this card
            else
              -- the person typed two different passwords
              show button "No match..."
              beep
              wait 4
              hide button "No match..."
              send opencard to this card
            end if
          end if
        end if
      end if
    else
      show button "Checking password database..."
      lock screen
      push card
      hide button "Checking password database..."
      -- go to check whether a card is set up for this person
      go to stack "Password stack"
      find whole PRESERVE_PWD in fld dwp
      if the result is empty then
        send mouseUp to bg button do_it
        set the lockMessages to false
      else
        set the lockMessages to true
        pop card
        show button "Access denied !!!"
        beep
        unlock screen
        set the lockMessages to false
        wait 4 seconds
        hide button "Access denied !!!"
        send opencard to this card
      end if
    end if
  end if
end returnkey

***** CARD #1, BUTTON #1: New Button *****
on mouseUp
  pop card
end mouseUp

***** CARD #2: PMS_driver_crd] *****
on returnkey
  global CHECK_DESIRE
  -- check if return key was not pressed after change password request
  if CHECK_DESIRE is not 9876 then
    -- release trap of returnkey
    pass returnkey
    -- but if it was...
  else
    -- set the msg box out of sight
    set the loc of msg to 22,-1500
    -- check if a password was typed
    if msg box = empty then
      -- none was typed, exit the process
      exit returnkey
      -- but if some password was typed...
    else
      -- store password typed
      put msg into TEMP_PWD
      put card fld abcd into PRESERVE_PWD
      -- empty the message box
      put empty into msg
      -- hide the message box
      hide msg
      -- restore the location of the message box to its original location
      set the loc of msg to 22,300
      -- check if the password given was the same as the one given at the
      -- entry level of the system
      -- if not...
      if TEMP_PWD is not PRESERVE_PWD then
        -- inform the user that wrong pwd
        beep
        show button "Access denied !!!"
        wait 4 seconds
        hide button "Access denied !!!"
        -- but if it was...
      else
        -- prompt the user for his new password
        ask "What is your new password ?"
        -- check if the user has typed something for password
        if it is not empty then
          -- store it temporarilly
          put it into PWD1
          -- prompt the user to type it again to make sure he remembers it
          ask "Type it again please..."
          -- check if the two passwords were the same
          if it = PWD1 then
            -- inform the user and change cursor to "wait"
            set cursor to 4
            show button "Setting password........"
            -- lock the screen to make things go faster
            lock screen
            -- prepare new screen after completion of the process
            hide button "Setting password........"
            -- preserve this card
            push card
            -- go to the pwd db to  find his card
            go to stack "Password stack"
            find whole PRESERVE_PWD in fld dwp
            -- set his pwd to his new pwd
            put PWD1 into fld dwp
            -- return to the driver card
            pop card
            -- restore screen
            unlock screen
          -- reenter to the PMS system with the new password or exit
          go first
          -- but if the person typed two different passwords
        else
          -- inform the user
          show button "No match........"
          -- get his attention
          beep
          -- wait a little so that he reads the message
          wait 4
          -- bring screen to its original condition
          hide button "No match........"
          -- ask the user if he wants to retry
          answer "Repeat the process ?" with "Ok" or "Cancel"
          -- if he does...
          if it is "Ok" then
            -- retry...
            send returnkey to this card
            -- but if not
          else
            -- exit the PMS system
            go first
          end if -- user wants to retry
        end if -- the two supplied pwds match
      end if -- pwd not empty
    end if -- pwd supplied is not the same as the original
  end if -- pwd was not supplied
 end if -- return key was not pressed after password request
end returnkey

***** CARD #2, FIELD #1: rmv_person_list *****
on mousedown
  -- unlock the field in order to be able to find the selection
  set the locktext of me to false
  -- select the name of the person clicked
  select item 1 of line clickline() of me
  -- restore the previous condition of the field
  set locktext of me to true
  -- check whether the user clicked on a blank line to exit
  if the selection is empty then
    exit mouseDown
    -- but if not...
  else
    -- save the selected text and form question
    put the selection into SAVE_SELECT
    put "Remove:" & SAVE_SELECT into ASK_VAR
    -- click outside the field to deselect the person
    click at 128,157
    -- verify with the user that he realy wants to remove this person from the db
    answer ASK_VAR with "Ok" or "Cancel"
    -- check if the user answered Ok
    if it is "Ok" then
      -- lock the screen to make things go faster
      lock screen
      -- hide the list
      hide button "Click on desired person"
      hide me
      -- show to the user that the list has desapeared
      unlock screen
      -- change cursor so that the user Knows he has to wait a little
      set cursor to 4
      -- inform the user
```

```
      lock screen
      -- hide info
      hide button "Removing person........"
      -- preserve this card
      push card
      -- go to password stack to find and delete the card
      go to stack "Password Stack"
      -- find the desired card
      find SAVE_SELECT in fld hq_fld_rank_name
      -- delete this card
      doMenu "Delete Card"
      -- compact the stack to preserve space
      doMenu "Compact Stack"
      -- return back to PMS_driver
      pop card
      -- restore screen
      unlock screen
      -- if the user changed his mind...
    else
      -- lock the screen to make things go faster
      lock screen
      -- hide the list
      hide button "Click on desired person"
      hide me
    end if -- do delete
  end if -- the user have clicked on an empty line
end mousedown

***** CARD #2, FIELD #2: change_access_list *****
on mousedown
  -- unlock the field in order to be able to find the selection
  set the locktext of me to false
  -- select the name of the person clicked
  select item 1 of line clickline() of me
  -- restore the previous condition of the field
  set locktext of me to true
  -- check whether the user clicked on a blank line to exit
  if the selection is empty then
    exit mouseDown
    -- but if not...
  else
    -- save the selected text and form question
    put the selection into SAVE_SELECT
    put "Change:" & SAVE_SELECT into ASK_VAR
    -- click outside the field to deselect the person
    click at 128,157
    -- verify with the user that he realy wants to remove this person from the db
    answer ASK_VAR with "Ok" or "Cancel"
    -- check if the user answered Ok
    if it is "Ok" then
      -- lock the screen to make things go faster
      lock screen
      -- hide the list
      hide button "Click on desired person"
      hide me
      -- inform user
      show button "searching........"
      -- show changes
      unlock screen
      -- lock again to make things faster
      lock screen
      go to stack "Password Stack"
      -- find the desired card
      find SAVE_SELECT in fld bg_fld_rank_name
      -- restore screen
      unlock screen
      -- if the user changed his mind...
    else
      -- lock the screen to make things go faster
      lock screen
      -- hide the list
      hide button "Click on desired person"
      hide me
    end if -- do change
  end if -- the user have clicked on an empty line
end mousedown

***** CARD #2, FIELD #5: DIVLIST *****
on mouseup
  global wcvar, linestemp,divlisttemp
  put the number of lines of me into linestemp
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into wcvar
  --lock screen
  hide card fld divselect
  hide me
  --unlock screen
  send mouseup to bg button "MRC spot check"
end mouseup

***** CARD #2, FIELD #6: DIVLIST1 *****
on mouseup
  global wcvar, linestemp,divlisttemp
  put the number of lines of me into linestemp
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into wcvar
  --lock screen
  hide card fld divselect
  hide me
  --unlock screen
  send mouseup to bg button "MRC spot check"
end mouseup

***** CARD #2, BUTTON #1: New Button *****
on mouseDown
  -- set the menu acording to password
  global DRIV_ENGINEERING_MENU
  -- use current quarter
  global CURRENT_Q
  -- show apropriate menu
  get HPopupMenu(DRIV_ENGINEERING_MENU,0,128,348)
  -- check if item was selected
  -- if yes...
  if it is not zero then
    -- preserve this card
    push card
    -- extruct selection
    put item 1 of it into theline
    put item 2 of it into theitem
    -- find what the selection is...
    get item theitem of line theline of DRIV_ENGINEERING_MENU
    -- preserve it
    put it into WHAT_WC
    -- get today's date
    put the long date into WHAT_DAY
    -- lock the screen to let things roll faster
    lock screen
    -- find current quarter's stack
    go to stack CURRENT_Q
    -- if it is not there
    if the result is not empty then
      -- unlock the screen
      unlock screen
      -- inform the user
      answer "Quarter not scheduled..."
      -- get attention
      beep
      -- exit event
      exit mouseDown
      -- but if it is there...
    else
      -- unhighlight the button
      set the hilite of me to false
      -- find the desired work center
      go to card 1 of bg DA_model
      find WHAT_WC in fld bg fld WC_DA
      -- if PMS do not exist for this WC
      if the result is not empty then
        -- return to driver
        pop card
        -- get attention
        beep
        -- restore screen
        unlock screen
        -- inform user
        answer "No PMS for this Work Center !!!"
        wait 4 seconds
        -- exit event
        exit mouseDown
        -- but if there are PMS for this WC...
      else
        -- find today's date
        find WHAT_DAY
        -- click to cancel "find" box
        click at 363,49
        -- restore screen
        unlock screen
      end if
    end if
  end if
  -- exit event
end mouseDown

***** CARD #2, BUTTON #2: New Button *****
on mouseDown
  -- set the menu acording to password
  global DRIV_COMBAT_SYS_MENU
  -- use current quarter
  global CURRENT_Q
  -- show apropriate menu
  get HPopupMenu(DRIV_COMBAT_SYS_MENU,0,168,304)
  -- check if item was selected
  -- if yes...
  if it is not zero then
    -- preserve this card
  push card
  -- extruct selection
  put item 1 of it into theline
  put item 2 of it into theItem
  -- find what the selection is...
  get item theItem of line theline of DRIV_COMBAT_SYS_MENU
  -- preserve it
  put it into WHAT_WC
  -- get today's date
  put the long date into WHAT_DAY
  -- lock the screen to let things roll faster
  lock screen
  -- find current quarter's stack
  go to stack CURRENT_Q
  -- if it is not there
  if the result is not empty then
    -- unlock the screen
    unlock screen
    -- inform the user
    answer "Quarter not scheduled..."
    -- get attention
    beep
    -- exit event
    exit mouseDown
    -- but if it is there...
  else
    -- unhighlight the button
    set the highlight of me to false
    -- find the desired work center
    go to card 1 of bg DA_model
    find WHAT_WC in fld bg_fld_WC_DA
    -- if PMS do not exist for this WC
    if the result is not empty then
      -- return to driver
      pop card
      -- get attention
      beep
      -- restore screen
      unlock screen
      -- inform user
      answer "No PMS for this Work Center !!!"
      wait 4 seconds
      -- exit event
      exit mouseDown
      -- but if there are PMS for this WC...
    else
      -- find today's date
      find WHAT_DAY
      -- click to cancel "find" box
      click at 363,49
      -- restore screen
      unlock screen
    end if
  end if
end if
  -- exit event
end mouseDown

***** CARD #2, BUTTON #3: New Button *****
on mouseDown
  -- set the menu acording to password
  global DRIV_SHIP_CONTROL_MENU
```

```
-- show apropriate menu
get HPopupMenu(DRIV_SHIP_CONTROL_MENU,0,208,260)
-- check if item was selected
-- if yes...
if it is not zero then
  -- preserve this card
  push card
  -- extruct selection
  put item 1 of it into theline
  put item 2 of it into theItem
  -- find what the selection is...
  get item theItem of line theline of DRIV_SHIP_CONTROL_MENU
  -- preserve it
  put it into WHAT_WC
  -- get today's date
  put the long date into WHAT_DAY
  -- lock the screen to let things roll faster
  lock screen
  -- find current quarter's stack
  go to stack CURRENT_Q
  -- if it is not there
  if the result is not empty then
    -- unlock the screen
    unlock screen
    -- inform the user
    answer "Quarter not scheduled..."
    -- get attention
    beep
    -- exit event
    exit mouseDown
    -- but if it is there...
  else
    -- unhighlight the button
    set the highlight of me to false
    -- find the desired work center
    go to card 1 of bg DA_model
    find WHAT_WC in fld bg_fld_WC_DA
    -- if PMS do not exist for this WC
    if the result is not empty then
      -- return to driver
      pop card
      -- get attention
      beep
      -- restore screen
      unlock screen
      -- inform user
      answer "No PMS for this Work Center !!!"
      wait 4 seconds
      -- exit event
      exit mouseDown
      -- but if there are PMS for this WC...
    else
      -- find today's date
      find WHAT_DAY
      -- click to cancel "find" box
      click at 363,49
      -- restore screen
      unlock screen
    end if
  end if
end if
```

```
***** CARD #2, BUTTON #4: New Button *****
on mouseDown
  -- set the menu acording to password
  global DRIV_SUPPORT_MENU
  -- use current quarter
  global CURRENT_Q
  -- show aproprlate menu
  get HPopupMenu(DRIV_SUPPORT_MENU,0,248,216)
  -- check if item was selected
  -- if yes...
  if it is not zero then
    -- preserve this card
    push card
    -- extruct selection
    put item 1 of it into theline
    put item 2 of it into theItem
    -- find what the selection is...
    get item theItem of line theline of DRIV_SUPPORT_MENU
    -- preserve it
    put it into WHAT_WC
    -- get today's date
    put the long date into WHAT_DAY
    -- lock the screen to let things roll faster
    lock screen
    -- find current quarter's stack
    go to stack CURRENT_Q
    -- if it is not there
    if the result is not empty then
      -- unlock the screen
      unlock screen
      -- inform the user
      answer "Quarter not scheduled..."
      -- get attention
      beep
      -- exit event
      exit mouseDown
      -- but if it is there...
    else
      -- unhighlight the button
      set the highlight of me to false
      -- find the desired work center
      go to card 1 [illegible] DA_model
      find WHAT_WC in fld bg fld WC_DA
      -- if PMS do not exist for this WC
      if the result is not empty then
        -- return to driver
        pop card
        -- get attention
        beep
        -- restore screen
        unlock screen
        -- inform user
        answer "No PMS for this Work Center !!!"
        wait 4 seconds
        -- exit event
        exit mouseDown
        -- but if there are PMS for this WC...
      else
        -- find today's date
        find WHAT_DAY
        -- click to cancel "find" box
        click at 363,49
        -- restore screen
        unlock screen
        end if
      end if
    end if
    -- exit event
end mouseDown

***** CARD #2, BUTTON #5: crd_but_PMS *****
on mouseDown
  global CHECK_DESIRE
  -- use current quarter
  global CURRENT_Q
  -- set up the appropriate menu for this person
  global DRIV_GENERAL_MENU
  -- form the menu
  get HPopUpMenu(DRIV_GENERAL_MENU,0,188,65)
  -- if the mouse button was released with in the menu...
  if it is not zero then
    -- save the osition the mouse was released into variables
    put item 1 of it into theline
    put item 2 of it into theItem
    -- unhilite this button
    set the highlight of me to false
        -- if "Password" was selected from menu
    if theline = 1 then
      -- check which item of the submenu was selected
      -- if "Add new member" was selected...
      if theItem = 2 then
        -- change cursor so that the user Knows he has to wait a little
        set cursor to 4
        -- lock the screen so that opperation is faster
        lock screen
        -- preserve this card
        push card
        -- go to template stack to retrieve a new form
        go to card "password_crd_model" of stack templates
        -- get a copy
        doMenu "Copy Card"
        -- come back
        pop card
        -- place it after the current card of driver
        doMenu "Paste Card"
        -- finaly unlock the screen to enable user see the card
        unlock screen
      end if
      -- if "Remove a member" was selected...
      if theItem = 3 then
        -- change cursor so that the user Knows he has to wait a little
        set cursor to 4
        -- inform user
        show button "Preparing list........"
        -- lock the screen so that opperation is faster
        lock screen
        -- hide button, so that when back the user will not see it any more
        hide button "Preparing list........"
        -- preserve this card
        push card
        -- go to password atack to retrieve all members in db
        go to stack "Password stack"
        -- empty variable to store retrirved information
        put empty into TRANSFER_PEOPLE
        -- check all cards and retrieve name and position
        repeat with i = 2 to the number of cards
          -- get desired info
          get fld bg_fld_rank_name of card i
      -- store it temporarily
      put it into VAR1
      get fld bg_fld_position of card i
      put  VAR1 & ", (" & it & ")" & return after TRANSFER_PEOPLE
    end repeat
    -- come back
    pop card
    --put the retrieved information into the field
    put fieldsort(TRANSFER_PEOPLE,1) into card fld rmv_person_list
    -- show the button and the field
    show card fld rmv_person_list
    show button "Click on desired person"
    -- finaly unlock the screen to enable user see the card
    unlock screen
  end if
  -- if "Change member's access" was selected...
  if theItem = 4 then
    -- change cursor so that the user Knows he has to wait a little
    set cursor to 4
    -- inform user
    show button "Preparing list........"
    -- lock the screen so that opperation is faster
    lock screen
    -- hide button, so that when back the user will not see it any more
    hide button "Preparing list........"
    -- preserve this card
    push card
    -- go to password stack to retrieve all members in db
    go to stack "Password stack"
    -- empty variable to store retrirved information
    put empty into TRANSFER_PEOPLE
    -- check all cards and retrieve name and position
    repeat with i = 2 to the number of cards
      -- get desired info
      get fld bg_fld_rank_name of card i
      -- store it temporarily
      put it into VAR1
      get fld bg_fld position of card i
      put  VAR1 & ". (" & it & ")" & return after TRANSFER_PEOPLE
    end repeat
    -- come back
    pop card
    --put the retrieved information into the field
    put fieldsort(TRANSFER_PEOPLE.1) into card fld change_access_list
    -- show the button and the field
    show card fld change_access_list
    show button "Click on desired person"
    -- finaly unlock the screen to enable user see the card
    unlock screen
  end if
  -- if "Change my password" was selected...
  if theItem = 5 then
    global CHECK_DESIRE
    -- prompt the user to enter his old password
    play "PASSWORD PLEASE"
    --  set flag to activate the return key
    put 9876 into CHECK_DESIRE
  end if
end if -- password menu was selected
-- if "Schedule" was selected from menu
if theline = 2 then
  -- check which item of the submenu was selected
```

```
    -- change the cursor to let the user know he has to wait
    set cursor to 4
    -- Create a new set of cards for next quarter
    NewQuarter
  end if
end if -- "schedule" menu was selected
-- if "Reports and messages" was selected from menu
if theline = 3 then
  -- check which item of the submenu was selected
  -- if "% done diagram - 1 WC" was selected...
  if theItem = 2 then

    -- This handler generates a graph for each month of the quarter specified
    -- for a particular WC
    -- create and clear three variables to work with
    global WORK_CENTERS_TO_REPORT
    global START_MONTH
    global TEMP_YEAR
    global GO_Q
    push card
    show bg button "Clearing old reports..."
    lock screen
    hide bg button "Clearing old reports..."
    go to stack compl_report
    repeat with i = 2 to the number of cards
      doMenu "Delete Card"
    end repeat
    pop card
    unlock screen
    -- ask the user for which WC he cares
    ask "WC to generate report for:"
    -- if the user changed his mind...
    if it is empty or it is "Cancel" then
      -- exit event
      exit mouseDown
      -- but if not
    else
      show bg button "Checking access privilege..."
      -- suspend screen drawing
      lock screen
      hide bg button "Checking access privilege..."
      -- temporarily store the WC
      put it into WORK_CENTERS_TO_REPORT
      -- change to upper
      repeat with in_char = 1 to length(WORK_CENTERS_TO_REPORT)
        if charToNum(char in_char of WORK_CENTERS_TO_REPORT) > 96 and
        charToNum(char in_char of WORK_CENTERS_TO_REPORT) < 123 then
          put numToChar(charToNum(char in_char of WORK_CENTERS_TO_REPORT) - 32) into

          char in_char of WORK_CENTERS_TO_REPORT
        end if
      end repeat
      -- get persons pwd
      put card fld abcd into CHECK_ACCESS
      -- preserve this card
      push card
      -- go to check if the WC entered is valid
      go to stack ship_data
      -- try to find it
      find WORK_CENTERS_TO_REPORT in fld wc_resp
      -- if failed...
      -- restore screen
      unlock screen
      -- inform the user...
      answer "No satch WC..."
      -- but if succeeded...
    else
      -- go to check if the user has access to this wc
      go to stack "password stack"
      -- look for his card
      find CHECK_ACCESS in fld dwp
      -- check if he has access to this wc
      if the hilite of button WORK_CENTERS_TO_REPORT then
        -- if he does...return to driver
        pop card
        -- restore screen
        unlock screen
        -- ask him what quarter he cares about
        put the date into MAKEDATE
        convert MAKEDATE to dateItems
        put item 2 of MAKEDATE & "/" & item 1 of MAKEDATE into MAKEDATE
        ask "Month in desired quarter (mm/yyyy)..." with MAKEDATE
        -- if he changed his mind
        if it is empty or it is "Cancel" then
          -- exit event
          exit mouseDown
          -- but if he typed something in
        else
          -- suspend screen drawing
          lock screen
          -- preserve card
          push card
          --store the month typed in temporarily
          put it into TEMP
          --split the month from the year
          put char 1 to 2 of TEMP into TEMP_MONTH
          if char 1 of TEMP_MONTH = "0" then
            delete char 1 of TEMP_MONTH
          end if
          put char 4 to 7 of TEMP into TEMP_YEAR
          -- for desired quarter
          if TEMP_MONTH = 1 or TEMP_MONTH = 2 or TEMP_MONTH = 3 then
            put "DA JAN-MAR" && TEMP_YEAR into GO_Q
            put 1 into START_MONTH
          end if
          if TEMP_MONTH = 4 or TEMP_MONTH = 5 or TEMP_MONTH = 6 then
            put "DA APR-JUN" && TEMP_YEAR into GO_Q
            put 4 into START_MONTH
          end if
          if TEMP_MONTH = 7 or TEMP_MONTH = 8 or TEMP_MONTH = 9 then
            put "DA JUL-SEP" && TEMP_YEAR into GO_Q
            put 7 into START_MONTH
          end if
          if TEMP_MONTH = 10 or TEMP_MONTH = 11 or TEMP_MONTH = 12 then
            put "DA OCT-DEC" && TEMP_YEAR into GO_Q
            put 10 into START_MONTH
          end if
          -- go to desired stack
          go to stack GO_Q
          -- if the stack is not in the system
          if the result is not empty then
            -- return to the driver
            pop card
            -- restore screen
            unlock screen
            -- inform the user
            answer "Quarter not in the system..."
            -- exit event
            exit mouseDown
            -- but if it is there...
          else
            pop card
            lock screen
            show button output_border
            show button "Select output device"
            show button to_printer
            show button to_screen
            show button dont_do_report
            unlock screen
          end if -- the result is not empty
        end if -- month is empty
      else
        -- return to driver
        pop card
        -- restore screen
        unlock screen
        -- get attention
        beep
        -- inform user
        show button "Access denied !!!"
        -- wait to let user see it
        wait 4 seconds
        -- exit event
        exit mouseDown
      end if -- the hilite of WC
    end if -- the result is not empty
  end if -- empty or cancel
end if
-- if "% done diagram - all WC" was selected...
if theItem = 3 then
  -- This handler generates a graph for each month of the quarter specified
  -- for all WC
  -- create and clear three variables to work with
  global WORK_CENTERS_TO_REPORT
  global START_MONTH
  global TEMP_YEAR
  global GO_Q
  push card
  show bg button "Clearing old reports..."
  lock screen
  hide bg button "Clearing old reports..."
  go to stack compl_report
  repeat with i = 2 to the number of cards
    doMenu "Delete Card"
  end repeat
  pop card
  unlock screen
  -- ask him what quarter he cares about
  put the date into MAKEDATE
  convert MAKEDATE to dateItems
  put item 2 of MAKEDATE & "/" & item 1 of MAKEDATE into MAKEDATE
  ask "Month in desired quarter (mm/yyyy)..." with MAKEDATE
  -- if he changed his mind
  if it is empty or it is "Cancel" then
    -- exit event
    exit mouseDown
    -- but if he typed something in
  else
    -- suspend screen drawing
    lock screen
    -- preserve card
    push card
    --store the month typed in temporarily
    put it into TEMP
    --split the month from the year
    put char 1 to 2 of TEMP into TEMP_MONTH
    if char 1 of TEMP_MONTH = "0" then
      delete char 1 of TEMP_MONTH
    end if
    put char 4 to 7 of TEMP into TEMP_YEAR
    -- for desired quarter
    if TEMP_MONTN = 1 or TEMP_MONTH = 2 or TEMP_MONTH = 3 then
      put "DA JAN-MAR" && TEMP_YEAR into GO_O
      put 1 into START_MONTH
    end if
    if TEMP_MONTH = 4 or TEMP_MONTH = 5 or TEMP_MONTH = 6 then
      put "DA APR-JUN" && TEMP_YEAR into GO_O
      put 4 into START_MONTH
    end if
    if TEMP_MONTH = 7 or TEMP_MONTH = 8 or TEMP_MONTH = 9 then
      put "DA JUL-SEP" && TEMP_YEAR into GO_O
      put 7 into START_MONTH
    end if
    if TEMP_MONTN = 10 or TEMP_MONTH = 11 or TEMP_MONTH = 12 then
      put "DA OCT-DEC" && TEMP_YEAR into GO_O
      put 10 into START_MONTH
    end if
    -- go to desired stack
    go to stack GO_O
    -- if the stack is not in the system
    if the result is not empty then
      -- return to the driver
      pop card
      -- restore screen
      unlock screen
      -- inform the user
      answer "Quarter not in the system..."
      -- exit event
      exit mouseDown
      -- but if it is there...
    else
      put empty into WORK_CENTERS_TO_REPORT
      go to card 1 of bg responsibility_bg of stack ship_data
      repeat with ii = 1 to the number of cards of bg responsibility_bg
        get fld wc_resp of card ii of bg responsibility_bg
        put it & "," after WORK_CENTERS_TO_REPORT
      end repeat
      pop card
      lock screen
      show button output_border
      show button "Select output device"
      show button to_printer
      show button to_screen
      show button dont_do_report
      unlock screen
    end if -- the result is not empty
  end if -- month is empty
end if
-- if "Bar coded label" was selected...
if theItem = 4 then
    lock screen
    -- preserve this card
    push card
    -- go to template stack to retrieve a new form
    go to card "bar_coded_model" of stack templates
    -- get a copy
    doMenu "Copy Card"
    -- come back
    pop card
    -- place it after the current card of driver
    doMenu "Paste Card"
    -- put todays date
    put the date into fld date
    -- finaly unlock the screen to enable user see the card
    unlock screen
    -- prepare for typing
    click at 374,91
  end if

  -- if "Navy message" was selected...
  if theItem = 5 then
  end if

  -- if "Ext. rep req" was selected...
  if theItem = 6 then
  end if

end if -- "Reports and messages" menu was selected
 -- if "Spot checks" was selected from menu
if theLine = 4 then
  -- check which item of the submenu was selected
  -- if "For entire ship" was selected...
  if theItem = 2 then
    sp_all_wc
  end if
  -- if "For one WC" was selected...
  if theItem = 3 then
    sp_1_wc
  end if
end if -- "Spot checks" menu was selected
 -- if the "Go" was selected from menu
if theLine = 5 then
  -- check which item of the submenu was selected
  -- if "MRC card" was selected...
  if theItem = 2 then
    ask "MRC number to go..."
    if it is empty or it is "Cancel" then
      exit mouseDown
    else
      put it into MRC_TO_FIND
      set cursor to 4
      push card
      lock screen
      go to stack PMS
      go to card (item 2 of line binary(mrc_list,MRC_TO_FIND) of fld
      mrc_list)
      unlock screen
    end if
  end if
  -- if "MIP card" was selected...
  if theItem = 3 then
```

```
    else
      put it into MIP_TO_FIND
      set cursor to 4
      push card
      lock screen
      go to stack PMS
      go to card (item 2 of line binary(mip_list.MIP_TO_FIND) of fld
      mip_list)
      unlock screen
    end if
  end if
  -- if "Log card" was selected...
  if theItem = 4 then
    ask "Unit name to go..."
    if it is empty or it is "Cancel" then
      exit mouseDown
    else
      lock screen
      put it into LOG_TO_FIND
      set cursor to 4
      push card
      go to stack log
      find LOG_TO_FIND in fld unit
      if the result is not empty then
        -- inform the user that no satch LOG card exists
        -- return to driver
        pop card
        unlock screen
        -- get attention
        beep
        show button "No satch LOG card........"
        wait 4 seconds
        -- restore screen
        hide button "No satch LOG card........"
        -- ...and exit event
        exit mouseDown
      else
        unlock screen
      end if
    end if
  end if
end if -- "Go" menu was selected
    -- if the "Print" was selected from menu
if theline = 6 then
  -- check which item of the submenu was selected
  -- if "Today'a OA cards - all" was selected...
  if theItem = 2 then
    lock screen
    push card
    -- go to stack ship_data
    -- put empty into HOLDER
    -- repeat with i = 1 to the number of cards of bg responsibility_bg
    --   put fld WC_resp & "," after HOLDER
    -- end repeat
    put "EM01,EM02,EM03," into HOLDER
    --pop card
    put the date into TODAY
    convert TOOAY to long date
    repeat with i = 1 to the number of items of HOLDER
      go to stack CUPRENT_O
      find item i of HOLDER
      find TODAY
      send mouseUp to bg button Print_OA
    end repeat
```

```
end if
-- if "Today's DA card - I WC" was selected...
if theItem = 1 then
  push card
  ask "DA card for which Work Center ?"
  if it is "Cancel" then
    exit mouseDown
  else
    put it into WC_to_print
    lock screen
    put card fld abcd into CHECK
    go to stack "Password stack"
    find CHECK in fld dwp
    if not the hilite of button WC_to_print then
      pop card
      unlock screen
      beep
      show button "Access denied !!!"
      wait 4 seconds
      hide button "Access denied !!!"
      exit mouseDown
    else
      go to stack CURRENT_O
      find WC to print
      if the result is not empty then
        pop card
        unlock screen
        beep
        answer "No such WC !!!"
        exit mouseDown
      else
        send mouseUp to bg button print_DA
        unlock screen
        pop card
      end if
    end if
  end if
end if
-- if "MIP card" was selected...
if theItem = 4 then
  ask "MIP number to print..."
  if it is empty or it is "Cancel" then
    exit mouseDown
  else
    put it into MIP_TO_FIND
    set cursor to 4
    push card
    lock screen
    go to card MIP_TO_FIND of stack PMS
    if the result is not empty then
      -- inform the user that no satch MIP
      -- return to driver
      pop card
      unlock screen
      -- get attention
      beep
      show button "No satch MIP........"
      wait 4 seconds
      -- restore the screen
      hide button "No satch MIP........"
      -- ...and exit event
      exit mouseDown
    else
      send mouseUp to bg button "Print this mip"
      pop card
      unlock screen
    end if
  end if
end if
-- if "MRC card" was selected...
if theItem = 5 then
  ask "MRC number to print..."
  if it is empty or it is "Cancel" then
    exit mouseDown
  else
    put it into MRC_TO_FIND
    set cursor to 4
    push card
    lock screen
    go to card MRC_TO_FIND of stack PMS
    if the result is not empty then
      -- inform the user that no satch MIP
      -- return to driver
      pop card
      unlock screen
      -- get attention
      beep
      show button "No satch MRC........"
      wait 4 seconds
      -- restore the screen
      hide button "No satch MRC........"
      -- ...and exit event
      exit mouseDown
    else
      send mouseUp to bg button "Print_mrc"
      pop card
      unlock screen
    end if
  end if
end if
-- if "Log card" was selected...
if theItem = 6 then
  ask "Unit name to print..."
  if it is empty or it is "Cancel" then
    exit mouseDown
  else
    lock screen
    put it into LOG_TO_FIND
    set cursor to 4
    push card
    go to stack log
    find LOG_TO_FIND in fld unit
    if the result is not empty then
      -- inform the user thst no satch LOG card
      -- return to driver
      pop card
      unlock screen
      -- get attention
      beep
      show button "No satch LOG card........"
      wait 4 seconds
      -- restore the screen
      hide button "No satch LOG card........"
      -- ...and exit event
      exit mouseDown
    else
      unlock screen
```

```
        or "Yes"
        if it is "No" then
          answer "Find the desired card and print..."
          with "Cancel" or "Ok"
          if it is "Cancel" then
            pop card
          else
            exit mouseDown
          end if
          send mouseUp to bg button print_log_rep
          pop card
        end if
      end if
    end if
  end if
end if -- "Print" menu was selected
-- if the "Toolbox" was selected from menu
if theline = 7 then
  -- check which item of the submenu was selected
  -- if "Add more MIP cards" was selected...
  if theItem = 2 then
  end if
  -- if "Convert MIP-MRC file to stack" was selected...
  if theItem = 3 then
  end if
  -- if "Empty PMS database" was selected...
  if theItem = 4 then
    push card
    go to card empty_pms_db of stack "PMS toolbox"
  end if
  -- if "MIP-MRC cards in PMS database" was selected...
  if theItem = 5 then
    push card
    go to card mip_mrc_list of stack "PMS toolbox"
  end if
  -- if "Delete a MIP set" was selected...
  if theItem = 6 then
    push card
    go to card del_mip_set of stack "PMS toolbox"
  end if
  -- if "WC responsibility" was selected...
  if theItem = 7 then
    push card
    go to card assign_wc_to_mrc of stack "PMS toolbox"
  end if
end if -- "Toolbox" menu was selected
    -- if the "Equipment log" was selected from menu
if theline = 8 then
  -- check which item of the submenu was selected
  -- if "Add entry" was selected...
  if theItem = 2 then
    -- change cursor so that the user Knows he has to wait a little
    set cursor to 4
    -- lock the screen so that opperation is faster
    lock screen
    -- preserve this card
    push card .
    -- go to template stack to retrieve a new form
    go to card "log_model" of stack templates
    -- get a copy
    doMenu "Copy Card"
    -- come back
    pop card
```

```
      doMenu "Paste Card"
      -- finaly unlock the screen to enable user see the card
      unlock screen
    end if
    -- if "Del entry" was selected...
    if theItem = 3 then
      -- ask the user for some kind of search string
      ask "Equipment identification to remove..."
      -- if the user asks for nothing or changes his mind
      if it is empty or it is "Cancel" then
        -- exit event
        exit mouseDown
        -- but if he wants to remove a card
      else
        -- preserve this card
        push card
        -- suspend screen drawing
        lock screen
        -- go to the log stack
        go to stack log
        -- find the requested string
        find it
        -- if not found
        if the result is not empty then
          -- go back
          pop card
          -- restore screen
          unlock screen
          -- get attention
          beep
          -- inform the user
          answer "Identification not found..."
          -- but if found...
        else
          --  show card found to the user
          unlock screen
          -- make sure you do not delete the wrong card
          answer "Delete this card ?" with "Ok" or "Cancel"
          -- if this is the card to be deleted...
          if it is "OK" then
            -- delete it
            doMenu "Delete Card"
            -- go back
            pop card
          else
            -- but if this card is not the correct one...
            -- ask the user to find the desired card
            answer "Do you want to search manually ?" with "No" or "Yes"
            -- if he doesn't want to...
            if it is "No" then
              -- go back
              pop card
            else
              -- but if he wants...
              show bg button delete_me
              -- exit event
              exit mouseDown
            end if
          end if
        end if
      end if
    end if
  end if -- "Equipment Log" menu was selected
end if -- mouse clicked with in menu
end mouseDown

***** CARD #2, BUTTON #6: Preparing New Calendar....... *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #8: Preparing list........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #9: Removing person........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #10: searching........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #11: Setting password........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #12: No match........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #13: Access denied !!! *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #14: No match MRC........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #15: No match MIP........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #16: No match LOG card........ *****
on mouseUp
  hide me
end mouseUp

***** CARD #2, BUTTON #19: to_printer *****
on mouseUp
  global PRINT_ME
  lock screen
  hide button output_border
  hide button to_screen
  hide button dont_do_report
  hide button "Select output device"
  hide me
  unlock screen
  put TRUE into PRINT_ME
  set cursor to 4
  show bg button "Preparing data..."
  lock screen
  hide bg button "Preparing data..."
  do_report
end mouseUp

***** CARD #2, BUTTON #20: to_screen *****
on mouseUp
  global PRINT_ME
  lock screen
  hide button output_border
  hide button to_printer
  hide button dont_do_report
  hide button "Select output device"
  hide me
  unlock screen
  put FALSE into PRINT_ME
  set cursor to 4
  show bg button "Preparing data..."
  lock screen
  hide bg button "Preparing data..."
  do_report
end mouseUp

***** CARD #2, BUTTON #21: dont_do_report *****
on mouseUp
  lock screen
  hide button output_border
  hide button to_screen
  hide button to_printer
  hide button "Select output device"
  hide me
  unlock screen
end mouseUp


SCRIPTS FOR STACK: pms
_____________________________________________________

****** STACK SCRIPT *****
on enterkey
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  if SEARCH_STRING_VAR is not empty then
    --go next
    if SEARCH_FIELD_VAR = "global" then
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    else
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    end if
  end if
end enterkey

function binary fieldname, value
  put 0 into start
```

```
    put trunc((finish - start)/2 + start) into n
    if value = item 1 of line n of fld fieldname then return n
    if value > item 1 of line n of fld fieldname then
      put n into start
    else
      put n into finish
    end if
  end repeat
  if value = item 1 of line finish of fld fieldname then
    return finish
  else
    return "not found"
  end if
end binary

***** BACKGROUND #1: list_bg *****
-- This function takes the contents of a container like a field and
-- sorts them acording to some item specified.  Arguments are the field
-- and the desired item to be sorted on.
-- The sorting is done in a ascending order.
function fieldsortnormal thefield, sort_on_item
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield - 1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines in thefield
      if item sort_on_item of line current of thefield < item
      sort_on_item of line leastlinenum of thefield then
        put current into leastlinenum
      end if
    end repeat
    if leastlinenum > primary then
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsortnormal

 ***** BKGND #1, BUTTON #1: New Button *****
on mouseUp
  doMenu "Print Report..."
end mouseUp

 ***** BKGND #1, BUTTON #2: LightHouse *****
on mouseUp
  go to card 1 of stack PMS_driver
end mouseUp

 ***** BKGND #1, BUTTON #3: generate *****
on mouseUp
  answer "Regenerate the list ?" with "Ok" or "Cancel"
  if it is "Ok" then
    set cursor to 4
    put empty into fld MIP list
```

```
      get the short name of card 1 of bg "MIP-1st page"
      put it into temp_name
      get the number of card 1 of bg "MIP-1st page"
      put temp_name &"," & it & return after fld MIP_list
    end repeat
    put fieldsortnormal (fld mip_list,1) into fld mip_list
    repeat with i = 2 to the number of cards of bg "MRC - 1st page."
      get the short name of card 1 of bg "MRC - 1st page."
      put it into temp_name
      get the number of card 1 of bg "MRC - 1st page."
      put temp_name &"," & it & return after fld MRC_list
    end repeat
    put fieldsortnormal (fld mrc_list,1) into fld mrc_list
  end if
end mouseUp

 ***** BKGND #2. BUTTON #2: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #2, BUTTON #4: New Button *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #2, BUTTON #5: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #2, BUTTON #6: d_field *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  subtract 5 from it
  if it < "0" then
    put "0" into it
  end if
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #2, BUTTON #7: d_field *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  add 5 to it
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #2, BUTTON #8: u_field *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  add 5 to it
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #2, BUTTON #9: print this mip *****
on mouseUp
  set cursor to watch
  lock screen
  put field ship_system into ship_temp
  put field pubs into pubs_temp
  put field config into config_temp
  put  field card_date into date_temp
  put field MIP into MIP_temp
  put field card_name into name_temp
  put 0 into GLOBAL_PAGES
  put 0 into GLOBAL_TOTAL_PAGES
  push card
  go next card
  put field test into test_temp
  put field MRC into MRC_temp
  put field MRCdescript into descr_temp
  put field period into period_temp
  put field rates into rates_temp
  put field man into man_temp
  put field relman into relman_temp
  --calculate pages required
  put 0 into FULL_PAGES
  put the number of lines of fld MRCdescript into PROCEDURE_LINES
  put (PROCEDURE_LINES - 27) div 44 + 1 into FULL_PAGES
  put (PROCEDURE_LINES - 27) mod 44 into REMAINING
  if REMAINING > 0 then
    put FULL_PAGES + 1 into FULL_PAGES
  end if
  put FULL_PAGES into GLOBAL_TOTAL_PAGES
  go to stack MIP_printed
  repeat with i=1 to(the number of cards - 1)
    go to card 2
    domenu "DELETE CARD"
  end repeat
  go to card "print_page_1" of stack "templates"
  doMenu "NEW CARD"
  doMenu "CUT CARD"
  go to stack MIP_printed
  doMenu "LAST"
  doMenu "PASTE CARD"
  put  ship_temp into field ship_system
  put  pubs_temp into field pubs
  put  config_temp into field config
  put  date_temp into field card_date
  put  MIP_temp into  field MIP
  put  name_temp into field card_name
  put 1 into count
  repeat while count < 3
    put line count of test_temp & return after field test
    put line count of MRC_temp & return after field MRC
    put line count of descr_temp & return after field MRCdescript
    put  line count of period_temp  & return after field period
    put line count of rates_temp  & return after field rates
    put line count of man_temp  & return after field man
    put line count of relman_temp & return after field relman
    put count + 1 into count
  end repeat
  go to card "print_page_2" of stack "templates"
  doMenu "NEW CARD"
  doMenu "CUT CARD"
  go to stack MIP_printed
  doMenu "LAST"
  doMenu "PASTE CARD"
  repeat while count < "28"
    put line count of test_temp & return after field test
    put line count of MRC_temp & return after field MRC
    put line count of descr_temp & return after field MRCdescript
    put  line count of period_temp  & return after field period
    put line count of rates_temp  & return after field rates
    put line count of man_temp  & return after field man
    put line count of relman_temp & return after field relman

    put count + 1 into count
  end repeat
  put the date into fld date_printed_pr
  put 1 into GLOBAL_PAGES
  put GLOBAL_PAGES into fld page_no_pr
  put GLOBAL_TOTAL_PAGES into fld page_of_pr
  put  PROCEDURE_LINES - "27" into  PROCEDURE_LINES
   put 47 into c_var
  put 72 into c_var2
  repeat until PROCEDURE_LINES <= 0
    go to card "print_page_3" of stack "templates"
    doMenu "NEW CARD"
    doMenu "CUT CARD"
    go to stack MIP_printed
    doMenu "LAST"
    doMenu "PASTE CARD"
    put  MIP_temp into  field MIP
    repeat while count < c_var
      put line count of test_temp & return after field test
      put line count of MRC_temp & return after field MRC
      put line count of descr_temp & return after field MRCdescript
      put  line count of period_temp  & return after field period
      put line count of rates_temp  & return after field rates
      put line count of man_temp  & return after field man
      put line count of relman_temp & return after field relman
      put count + 1 into count
    end repeat
    put PROCEDURE_LINES - 19 into PROCEDURE_LINES
    go to card "print_page_2" of stack "templates"
    doMenu "NEW CARD"
    doMenu "CUT CARD"
    go to stack MIP_printed
    doMenu "LAST"
    doMenu "PASTE CARD"
    repeat while count < c_var2
      put line count of test_temp & return after field test
      put line count of MRC_temp & return after field MRC
      put line count of descr_temp & return after field MRCdescript
      put  line count of period_temp  & return after field period
```

```
    end repeat
    put PROCEDURE_LINES - "25" into PROCEDURE_LINES
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
    put c_var + "44" into c_var
    put c_var2 + "44" into c_var2
  end repeat
  put the number of cards  into c_total
  go to card 1 of stack MIP_printed
  domenu "DELETE CARD"
  print all cards
  doMenu "Compact Stack"
  pop card
  unlock screen
end mouseUp

 ***** BKGND #2, BUTTON #10: append *****
on mouseUp
  global DEST
  set cursor to 4
  push card
  push card
  go to stack DEST
  put the number of cards into HOW_MANY_CARDS
  pop card
  go last
  repeat with i = 1 to HOW_MANY_CARDS
    put "Processing MIP :" && DEST && "                    card" && i && "out of" &&
    HOW_MANY_CARDS into msg
    push card --last card of PMS stack
    go to card 1 of stack DEST
    if the short name of this bg is "MIP-1st page" then
      go to stack PMS
      go to first card of bg "MIP-1st page"
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "card_date" into VAR1
      put field "pubs" into VAR2
      put field "config" into VAR3
      put field "MIP" into VAR4
      put field "card_NAME" into VAR5
      put field "ship_system" into VAR6
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "card_date"
      put VAR2 into field "pubs"
      put VAR3 into field "config"
      put VAR4 into field "MIP"
      put VAR5 into field "card_NAME"
      put VAR6 into field "ship_system"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "mip 2nd page" then
      go to stack PMS
      push card
      go to card 1 of stack DEST
      put field "MIP" into VAR1
      put field "test" into VAR2
      put field "MRC" into VAR3
      put field "MRCdescript" into VAR4
      put field "period" into VAR5
      put field "rates" into VAR6
      put field "man" into VAR9
      put field "relman" into VAR10
      put field "card_NAME" into VAR11
      put field "where_am_I" into VAR12
      put field "MRC_of_this_MIP_active" into VAR13
      put field "MRC_of_this_MIP_inactive" into VAR14
      -- put field "data_MRC" into VAR15
      -- put field "in_data_MRC" into VAR16
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MIP"
      put VAR2 into field "test"
      put VAR3 into field "MRC"
      put VAR4 into field "MRCdescript"
      put VAR5 into field "period"
      put VAR6 into field "rates"
      put VAR9 into field "man"
      put VAR10 into field "relman"
      put VAR11 into field "card_NAME"
      put VAR12 into field "where_am_i"
      put VAR13 into field "MRC_of_this_MIP_active"
      put VAR14 into field "MRC_of_this_MIP_inactive"
      -- put VAR15 into field "data_MRC"
      -- put VAR16 into field "in_data_MRC"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "MRC - 1st page." then
      go to stack PMS
      go to first card of bg "MRC - 1st page."
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "MRCno" into VAR1
      put field "what_equipment" into VAR2
      put field "MN" into VAR3
      put field "elapsed" into VAR4
      put field "mrcode" into VAR5
      put field "periodicity" into VAR6
      put field "date_edited" into VAR9
      put field "rel_maint" into VAR11
      put field "mr_description_field" into VAR12
      put field "rates_mh_field" into VAR13
      put field "equipment_field" into VAR14
      put field "subsystem_field" into VAR15
      put field "data" into VAR16
      put field "ship_system_field" into VAR17
      put field "system_field" into VAR18
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MRCno"
      put VAR2 into field "what_equipment"
      put VAR3 into field "MH"
      put VAR4 into field "elapsed"
      put VAR5 into field "mrcode"
      put VAR6 into field "periodicity"
      put VAR9 into field "date_edited"
      put VAR11 into field "rel_maint"
      put VAR12 into field "mr_description_field"
      put VAR13 into field "rates_mh_field"
      put VAR14 into field "equipment_field"
      put VAR15 into field "subsystem_field"
      put VAR16 into field "data"
      put VAR17 into field "ship_system_field"
      put VAR18 into field "system_field"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "MRC 2nd page" then
      go to stack PMS
      go to first card of bg "MRC 2nd page"
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "MRCno" into VAR1
      put field "what_equipment" into VAR2
      put field "procedure" into VAR3
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MRCno"
      put VAR2 into field "what_equipment"
      put VAR3 into field "procedure"
      set the name of this card to NAME_VAR
      next repeat
    end if
  end repeat
  pop card
end mouseUp

 ***** BKGND #2, BUTTON #11: New Button *****
on mouseUp
  push card
  lock screen
  go next
  go next
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #2, BUTTON #12: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #2, BUTTON #13: New Button *****
on mouseUp
  ask "Which MIP do you wish to view ?"
  if it is empty or it is "Cancel" then
```

```
    exit mouseup
  else
    lock screen
    push card
    go to card 1
    --put quote & it & quote into it
    put binary (mip_list,it) into it
    if it is not "not found" then
      put item 2 of line it of fld mip_list into it
      go to card it
    else
      pop card
      unlock screen
      answer ("Card not in the PMS database !!!")
    end if
  end if
end mouseUp

 ***** BKGND #3, FIELD #2: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                 "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "       " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #3: MRC *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                 "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "       " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #4: MRCdescript *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
```

```
  repeat until line it of field MRC <> "                 "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "       " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #5: period *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                 "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "       " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #6: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                 "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "       " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #7: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me)'+ it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                 "
  if "      " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, FIELD #8: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "      " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #3, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #3, BUTTON #3: New Button *****
on mouseUp
  go previous
  send mouseup to  bg button "print this mip"
end mouseUp

 ***** BKGND #3, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #3, BUTTON #14: New Button *****
on mouseUp
  visual effect scroll right
  pop card
end mouseUp

 ***** BKGND #3, BUTTON #16 *****
on mouseUp
  set the visible of fld MRC_of_this_MIP_active to not the visible of
  fld MRC_of_this_MIP_active
  set the visible of fld MRC_of_this_MIP_inactive to the visible of
  fld MRC_of_this_MIP_active
end mouseUp

 ***** BKGND #3, BUTTON #17: New Button *****
on mouseUp
  ask "Which MIP do you wish to view ?"
  if it is empty or it is "Cancel" then
    exit mouseup
  else
    lock screen
    push card
    go to card 1
    --put quote & it & quote into it
    put binary (mip_list,it) into it
    if it is not "not found" then
      put item 2 of line it of fld mip_list into it
      go to card it
    else
      pop card
      unlock screen
      answer ("Card not in the PMS database !!!")
    end if
  end if
end mouseUp

 ***** BKGND #3, BUTTON #18: New Button *****
on mouseUp
  push card
  lock screen
  go next
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #3, BUTTON #19: down *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field relman into it
  add 156 to it
  set the lockscreen to true
  set the scroll of field test to it
  set the scroll of field MRC to it
  set the scroll of field MRCdescript to it
  set the scroll of field period to it
  set the scroll of field rates to it
  set the scroll of field man to it
  set the scroll of field relman to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #3, BUTTON #20: left *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #3, BUTTON #21: go_top *****
on mouseDown
  set the lockscreen to true
  set the scroll of field test to 0
  set the scroll of field MRC to 0
  set the scroll of field MRCdescript to 0
  set the scroll of field period to 0
  set the scroll of field rates to 0
  set the scroll of field man to 0
  set the scroll of field relman to 0
  set the lockscreen to false
end mouseDown

 ***** BKGND #3, BUTTON #22: up *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field relman into it
  subtract 156 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of field test to it
  set the scroll of field MRC to it
  set the scroll of field MRCdescript to it
  set the scroll of field period to it
  set the scroll of field rates to it
  set the scroll of field man to it
  set the scroll of field relman to it
  set the lockscreen to false
end mouseDown

***** BACKGROUND #4: MRC - 1st page. *****
on do_select
  put "Add,Normal Duty,At Sea,Holiday,Overhaul,Inactive Status" into menu1
  put return & "Remove,Normal Duty,At Sea,Holiday,Overhaul,Inactive Status,This" after m
enu1
  get HPopupMenu(menu1,0,the mouseV,the mouseH)
  put 0 into countvisible
  put TRUE into CHECK
  repeat with i = 1 to 5
    if the visible of button i then put countvisible + 1 into countvisible
  end repeat
  if countvisible = 1 then put FALSE into CHECK
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theItem
    if theline = 1 and theItem = 2 then
      show button anchor
    end if
    if theline = 1 and theItem = 3 then
      show button at_sea
    end if
    if theline = 1 and theItem = 4 then
      show button flag
    end if
    if theline = 1 and theItem = 5 then
      show button overhaul
    end if
    if theline = 1 and theItem = 6 then
      show button inactive
    end if
    if theline = 2 and theItem = 2 and CHECK then
      hide button anchor
    end if
    if theline = 2 and theItem = 3 and CHECK then
    if theline = 2 and theItem = 4 and CHECK then
      hide button flag
    end if
    if theline = 2 and theItem = 5 and CHECK then
      hide button overhaul
    end if
    if theline = 2 and theItem = 6 and CHECK then
      hide button inactive
    end if
    if theline = 2 and theItem = 7 and CHECK then
      hide the target
    end if
    adjust
  end if
end do_select
on adjust
  put -5 into newloc
  repeat with i = 1 to 5
    if the visible of button i then
      put newloc + 36 into newloc
      set the loc of button i to newloc,39
    end if
  end repeat
end adjust

on enterkey
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  if SEARCH_STRING_VAR is not empty then
    --go next
    if SEARCH_FIELD_VAR = "global" then
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    else
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    end if
  end if
end enterkey

 ***** BKGND #4, BUTTON #1: print_mrc *****
on mouseUp
  set cursor to 4
  lock screen
  put 0 into GLOBAL_PAGES
  put 0 into GLOBAL_TOTAL_PAGES
  --clear destination
  put the id of this card into ID_COME_BACK
  push card
  go to stack MRC_print
  repeat with i = 1 to (the number of cards - 2)
    go to card 3
    doMenu "Delete Card"
  end repeat
  pop card     --Stack:empty
  put item 1 of line 4 of fld data + 1 into NUMBER_OF_PROCEDURE_PAGES
  put item 2 of line 4 of fld data into DIAGRAMS
  put empty into TRANSFER_PROCEDURE
  repeat with PROC_PAGES = 1 to NUMBER_OF_PROCEDURE_PAGES
    go next --2nd mrc card
    put fld procedure after TRANSFER_PROCEDURE
  end repeat
  repeat with PROC_LINES = 1 to the number of lines of TRANSFER_PROCEDURE
    if line PROC_LINES of TRANSFER_PROCEDURE ="-----------------------------------------------
-------------------------------------------------" then
      put "_____________________________________________________________________"
into
      line PROC_LINES of TRANSFER_PROCEDURE
    end if
  end repeat
  --calculate & make pages required
  put 0 into FULL_PAGES
  put the number of lines of TRANSFER_PROCEDURE into PROCEDURE_LINES
  if PROCEDURE_LINES > 25 then
    put ((PROCEDURE_LINES - 25) mod 47) into REMAINING_LINES
    put ((PROCEDURE_LINES - 25) div 47) into FULL_PAGES
    if REMAINING_LINES > 0 then put FULL_PAGES + 1 into FULL_PAGES
  end if
  repeat with i = 1 to FULL_PAGES
    go to card print_mrc_page3 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page2
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  repeat with i = 1 to DIAGRAMS
    go to card print_mrc_page4 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page7
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  put TABLES div 2 into TABLE_PAIRS
  put TABLES mod 2 into TABLE_REMAIN
  repeat with i = 1 to TABLE_PAIRS
    go to card print_mrc_page4 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page5
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  repeat with i = 1 to TABLE_REMAIN
    go to card print_mrc_page4 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
```

```
    doMenu "Paste Card"
    pop card  --Stack:click card,1st mrc card
    go to card print_mrc_page6
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  put (the number of cards / 2) into GLOBAL_TOTAL_PAGES
  pop card  --Stack:click card.  We are back to first MRC
  --make page 1
  put empty into WHAT_STATUS
  if the visible of button anchor then
    put "anchor," after WHAT_STATUS
  end if
  if the visible of button at_sea then put "at_sea," after WHAT_STATUS
  if the visible of button flag then put "flag," after WHAT_STATUS
  if the visible of button overhaul then put "overhaul," after WHAT_STATUS
  if the visible of button inactive then put "inactive," after WHAT_STATUS
  put fld what_equipment into VAR1
  put fld MRCno into VAR2
  put fld ship_system_field into VAR3
  put fld subsystem_field into VAR4
  put fld mrcode into VAR5
  put fld periodicity into VAR6
  put fld system_field into VAR7
  put fld equipment_field into VAR8
  put fld rates_mh_field into VAR9
  put fld mr_description field into VAR10
  put fld MH into VAR11
  put fld elapsed into VAR12
  put fld date_edited into VAR13
  push card   --Stack:click card,1st of mrc
  go to card 1 of stack MRC_print
  hide bg button anchor
  hide bg button at_sea
  hide bg button flag
  hide bg button overhaul
  hide bg button inactive
  if WHAT_STATUS contains "anchor" then show bg button anchor
  if WHAT_STATUS contains "at_sea" then show bg button at_sea
  if WHAT_STATUS contains "flag" then show bg button flag
  if WHAT_STATUS contains "overhaul" then show bg button overhaul
  if WHAT_STATUS contains "inactive" then show bg button inactive
  put -5 into newloc
  if the visible of bg button anchor then
    put newloc + 36 into newloc
    set the loc of bg button anchor to newloc,39
  end if
  if the visible of bg button at_sea then
    put newloc + 36 into newloc
    set the loc of bg button at_sea to newloc,39
  end if
  if the visible of bg button flag then
    put newloc + 36 into newloc
    set the loc of bg button flag to newloc,39
  end if
  if the visible of bg button overhaul then
    put newloc + 36 into newloc
    set the loc of bg button overhaul to newloc,39
  end if
  if the visible of bg button inactive then
    put newloc + 36 into newloc
    set the loc of bg button inactive to newloc,39
  end if
  put VAR1 into fld what_equipment_pr
  put VAR2 into fld MRCno_pr
  put VAR3 into fld ship_system_field_pr
  put VAR4 into fld subsystem_field_pr
  put VAR5 into fld mrcode_pr
  put VAR6 into fld periodicity_pr
  put VAR7 into fld system_field_pr
  put VAR8 into fld equipment_fld_pr
  put VAR9 into fld rates_mh_field_pr
  put VAR10 into fld mr_description_field_pr
  put VAR11 into fld MH_pr
  put VAR12 into fld elapsed_pr
  put VAR13 into fld date_edited_pr
  go to card 2 of stack MRC_print
  put 0 into START_LINE
  put 0 into LAST_LINE
  put LAST_LINE + 1 into START_LINE
  put START_LINE + 24 into LAST_LINE
  put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
  fld procedure_pr
  put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
  put the date into fld date_printed pr
  put 1 into GLOBAL_PAGES
  put GLOBAL_PAGES into fld page_no_pr
  put GLOBAL_TOTAL_PAGES into fld page_of_pr
  put the id of this card into TEMPID
  repeat until PROCEDURE_LINES <= 0
    go next
    put VAR2 into fld MRCno_pr
    put LAST_LINE + 1 into START_LINE
    put START_LINE + 21 into LAST_LINE
    put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
    fld procedure_pr
    put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
    go next
    put the id of this card into TEMPID
    put LAST_LINE + 1 into START_LINE
    put START_LINE + 24 into LAST_LINE
    put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
    fld procedure_pr
    put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
  end repeat
  pop card   --Stack:click card.  We are at 1st mrc
  repeat for NUMBER_OF_PROCEDURE_PAGES
    go next
  end repeat
  repeat with i = 1 to DIAGRAMS
    go next   --diagram card
    push card  --Stack:click card,diagram card
    put card field ledgent into TRANSFER_LEDGENT
    choose select tool
    drag from 23,74 to 481,281
    type "c" with commandkey
    go to stack MRC_print
    go to TEMPID
    go next
    type "v" with commandkey
    drag from 107,188 to 107,208
```

```
    go next
    put the id of this card into TEMPID
    put TRANSFER_LEDGENT into fld ledgent_pr
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
    pop card    --Stack:click card.  We are at diagram card
  end repeat
  repeat with i = 1 to TABLES
    go next   --table card
    push card  --Stack:click card,table card
    choose select tool
    drag from 23,74 to 481,281
    type "c" with commandkey
    go to stack MRC_print
    go to TEMPID
    go next
    put the id of this card into TEMPID
    type "v" with commandkey
    if i mod 2 = 0 then
      drag from 107,188 to 107,156
      put the date into fld date_printed_pr
      put GLOBAL_PAGES + 1 into GLOBAL_PAGES
      put GLOBAL_PAGES into fld page_no_pr
      put GLOBAL_TOTAL_PAGES into fld page_of_pr
    else
      drag from 107,188 to 107,208
      choose browse tool
      put VAR2 into fld MRCno_pr
    end if
    pop card    --Stack:click card.  We are at table card
  end repeat
  go to stack MRC_print
  go last
  if the short name of this card = "print_mrc_page6" then
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
  end if
  doMenu "Compact Stack"
  go first
  print all cards
  go to stack "PMS"
  go to IO_COME_BACK
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #2: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #4, BUTTON #3: to argos *****
on mouseUp
  push card
  lock screen
  if line 5 of fld data = "ARGOS" then
    pop card
    unlock screen
    answer "No conection to ARGOS has been established..."
    go to stack ARGOS
    go to card it
    unlock screen
  end if
end mouseUp

 ***** BKGND #4, BUTTON #4: New Button *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #4, BUTTON #5: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #4, BUTTON #6: New Button *****
on mouseUp
  set cursor to 4
  push card
  lock screen
  put line 1 of fld data into TEMPMIP
  delete char 1 to 4 of TEMPMIP
  go first
  go to card (item 2 of line binary(mip_list,TEMPMIP) of fld
  mip_list)
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #7: New Button *****
on mouseUp
  push card
  lock screen
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #4, BUTTON #8: open repair parts *****
on mouseUp
  lock screen
  show bg button "repair parts"
  show bg button close
  show card field "repair parts field"
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #9: New Button *****
on mouseUp
  push card
   -- This part between the asterisks sould be removed when real pictures
  -- will be entered to the system. Now a couple of demonstration pictures
  -- are provided as space holders in order to demonstrate the
  -- capability of the system.  The pictures were taken by the use of an
  -- ordinary 35 mm camera. The part of code after the  asterisks
  -- should be inserted here, and the "--" should be removed from the
  -- begining of the lines.
  visual effect dissolve
```

```
  go to stack movies
  -- repeat until fld page_no = 1
  --   go prev
  -- end repeat
  -- if line 6 of fld data = "MOVIE" then
  --   pop card
  --   unlock screen
  --   answer "No pictures available for this job..."
  -- else
  --   visual effect dissolve
  --   put field "MRCno" & "-movie" into it
  --   push card
  --   go to card it of stack movies
  -- end if
end mouseUp

 ***** BKGND #4, BUTTON #11: close *****
on mouseUp
  lock screen
  hide card field "repair parts field"
  hide me
  hide bg button "Repair Parts"
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #13: locker *****
on mouseDown
  put "Unlock Description" into menu1
  get HPopupMenu(menu1,0,193,298)
end mouseDown

on mouseUp
  hide me
  show bg button unlocker
  set the locktext of fld mr_description_field to false
end mouseUp


 ***** BKGND #4, BUTTON #16: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #4, BUTTON #17: forget_it *****
on mouseDown
  put return & "Exit Search" into menu
  get HPopupMenu (menu.0,125,355)
end mouseDown
on mouseUp
  lock screen
  show bg button search
  hide bg button "search for:"
  hide bg button search_menu
  hide bg button search_win
  hide bg button search_now
  hide fld search_by
  hide fld search_string
  hide me
  put empty into fld search_string
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #18: search_menu *****
```

```
on mouseDown
  global SEARCH_FIELD_VAR
  global DISPLAY
  put "MRC #" into menu
  put return & "Ship System" after menu
  put return & "Subsystem" after menu
  put return & "System" after menu
  put return & "Equipment" after menu
  put return & "MRC Code" after menu
  put return & "Periodicity" after menu
  put return & "Rates" after menu
  put return & "Total mh" after menu
  put return & "Description" after menu
  get HPopupMenu (menu,0,the mouseV - 40,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put line theline of menu into fld search_by
    if theline = 1 then
      put "MRCno" into SEARCH_FIELD_VAR
    end if
    if theline = 2 then
      put "ship_system_field" into SEARCH_FIELD_VAR
    end if
    if theline = 3 then
      put "subsystem_field" into SEARCH_FIELD_VAR
    end if
    if theline = 4 then
      put "system_field" into SEARCH_FIELD_VAR
    end if
    if theline = 5 then
      put "equipment_field" into SEARCH_FIELD_VAR
    end if
    if theline = 6 then
      put "mrcode" into SEARCH_FIELD_VAR
    end if
    if theline = 7 then
      put "periodicity" into SEARCH_FIELD_VAR
    end if
    if theline = 8 then
      put "rates_mh_field" into SEARCH_FIELD_VAR
    end if
    if theline = 9 then
      put "mh" into SEARCH_FIELD_VAR
    end if
    if theline = 10 then
      put "mr_description_field" into SEARCH_FIELD_VAR
    end if
    put line theline of menu into DISPLAY
  else
    put "Global Search" into fld search_by
    put "global" into SEARCH_FIELD_VAR
  end if
end mouseDown
on mouseUp
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #4, BUTTON #19: search_now *****
on mouseDown
  put return & "Find It" into menu
  get HPopupMenu (menu,0,125,335)
end mouseDown
```

```
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  push card
  set cursor to 4
  lock screen
  put fld search_string into SEARCH_STRING_VAR
  put length(SEARCH_STRING_VAR) into LAST_CHAR
  if char LAST_CHAR of SEARCH_STRING_VAR = return then
    put empty into char LAST_CHAR of SEARCH_STRING_VAR
  end if
  put empty into fld search_string
  send mouseUp to bg button forget_it
  if SEARCH_STRING_VAR is not empty then
    --go next
    if SEARCH_FIELD_VAR = "global" then
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        pop card
        beep
        answer "String not found..."
      end if
    else
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      if the result is not empty then
        pop card
        beep
        answer "String not found..."
      end if
    end if
  end if
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #21: search *****
on mouseUp
  global SEARCH_FIELD_VAR
  put "global" into SEARCH_FIELD_VAR
  lock screen
  put "Global Search" into fld search_by
  put empty into fld search_string
  show bg button search win
  show bg button search menu
  show bg button search now
  show bg button forget_it
  show bg button "search for:"
  show fld search_by
  show fld search_string
  hide me
  unlock screen
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #4, BUTTON #22: dt_opened *****
on mouseUp
  hide me
  show bg button dt_closed
  set the visible of fld data to not the visible of fld data
  set the visible of fld "rel_maint" to the visible of fld data
  set the visible of fld "other_wc_involved" to the visible of fld data
end mouseUp

 ***** BKGND #4, BUTTON #23: dt_closed *****
```

```
  show bg button dt_opened
  set the visible of fld data to not the visible of fld data
  set the visible of fld "rel_maint" to the visible of fld data
  set the visible of fld "other_wc_involved" to the visible of fld data
end mouseUp

***** BACKGROUND #5: MRC 2nd page *****
on opencard
  if fld page_no = fld of_pages then hide button go_next
  else
    show button go_next
  end if
end opencard

 ***** BKGND #5, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #5, BUTTON #3: New Button *****
on mouseUp
  push card
  repeat until fld page_no = 1
    go prev
  end repeat
  send mouseUp to bg button print_mrc
  pop card
end mouseUp

 ***** BKGND #5, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #5, BUTTON #5: to argos *****
on mouseUp
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  if line 5 of fld data = "ARGOS" then
    pop card
    unlock screen
    answer "No conection to ARGOS has been established..."
  else
    put line 5 of fld data into it
    go to stack ARGOS
    go to card it
    unlock screen
  end if
end mouseUp

 ***** BKGND #5, BUTTON #6: New Button *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #5, BUTTON #7: New Button *****
on mouseUp
  visual effect dissolve
  put field "MRC-no" & "-movie" into it
  push card
```

```
  go to card it
end mouseUp

 ***** BKGND #5, BUTTON #8: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #5, BUTTON #9: New Button *****
on mouseUp
  set cursor to 4
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  put line 1 of fld data into TEMPMIP
  delete char 1 to 4 of TEMPMIP
  go first
  go to card (item 2 of line binary(mip_list,TEMPMIP) of fld
  mip_list)
  unlock screen
end mouseUp

 ***** BKGND #5, BUTTON #10: New Button *****
on mouseUp
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #5, BUTTON #11: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #5, BUTTON #12: New Button *****
on mouseUp
  ask "WHICH MRC DO YOU WISH TO VIEW" with "MRC NUMBER"
  if (it is empty)  then
    exit mouseup
  else
    lock screen
    find whole it in field "MRC"
    if the result = empty then
      put the foundtext into dest
      push card
      go to card dest
      unlock screen
    else
      unlock screen
      exit mouseup
    end if
  end if
end mouseUp

***** BACKGROUND #6: MRC graphic page *****
on opencard
  if fld page_no = fld of_pages then hide button go_next
  else
    show button go_next
  end if
end opencard

***** BKGND #6, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

***** BKGND #6, BUTTON #3: New Button *****
on mouseUp
  push card
  repeat until fld page_no = 1
    go prev
  end repeat
  send mouseUp to bg button print_mrc
  pop card
end mouseUp

***** BKGND #6, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

***** BKGND #6, BUTTON #6: New Button *****
on mouseUp
  go prev
end mouseUp

***** BKGND #6, BUTTON #7: New Button *****
on mouseUp
  pop card
end mouseUp

***** BKGND #6, BUTTON #8: New Button *****
on mouseUp
  set cursor to 4
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  put line 1 of fld data into TEMPMIP
  delete char 1 to 4 of TEMPMIP
  go first
  go to card (item 2 of line binary(mip_list,TEMPMIP) of fld
  mip_list)
  unlock screen
end mouseUp

***** BKGND #6, BUTTON #9: New Button *****
on mouseUp
  push card
  lock screen
```

```
  end repeat
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

***** BKGND #6, BUTTON #10: New Button *****
on mouseUp
  visual effect dissolve
  put field "MRC-no" & "-movie" into it
  push card
  go to card it
end mouseUp

***** BKGND #6, BUTTON #11: New Button *****
on mouseUp
  ask "WHICN MRC DO YOU WISN TO VIEW" with "MRC NUMBER"
  if (it is empty)  then
    exit mouseup
  else
    lock screen
    find whole it in field "MRC"
    if the result = empty then
      put the foundtext into dest
      push card
      go to card dest
      unlock screen
    else
      unlock screen
      exit mouseup
    end if
  end if
end mouseUp

***** BACKGROUND #7: MRC table page *****
on opencard
  if fld page_no = fld of_pages then hide button go_next
  else
    show button go_next
  end if
end opencard

***** BKGND #7, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

***** BKGND #7, BUTTON #3: New Button *****
on mouseUp
  push card
  repeat until fld page_no = 1
    go prev
  end repeat
  send mouseUp to bg button print_mrc
```

```
***** BKGND #7, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

***** BKGND #7, BUTTON #6: New Button *****
on mouseUp
  go prev
end mouseUp

***** BKGND #7, BUTTON #7: New Button *****
on mouseUp
  pop card
end mouseUp

***** BKGND #7, BUTTON #8: New Button *****
on mouseUp
  set cursor to 4
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  put line 1 of fld data into TEMPMIP
  delete char 1 to 4 of TEMPMIP
  go first
  go to card (item 2 of line binary(mip list,TEMPMIP) of fld
mip_list)
  unlock screen
end mouseUp

***** BKGND #7, BUTTON #9: New Button *****
on mouseUp
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

***** BKGND #7, BUTTON #10: New Button *****
on mouseUp
  visual effect dissolve
  put field "MRC-no" & "-movie" into it
  push card
  go to card it
end mouseUp

***** BKGND #7, BUTTON #11: New Button *****
on mouseUp
  ask "WHICH MRC DO YOU WISH TO VIEW" with "MRC NUMBER"
  if (it is empty)  then
    exit mouseup
  else
    lock screen
    find whole it in field "MRC"
    if the result = empty then
      put the foundtext into dest
      push card
      go to card dest
      unlock screen
    else
      unlock screen
      exit mouseup
    end if
  end if
end mouseUp

***** CARD #4, BUTTON #1: anchor *****
on mouseDown
  do_select
end mouseDown

***** CARD #4, BUTTON #2: at_sea *****
on mouseDown
  do_select
end mouseDown

***** CARD #4, BUTTON #3: flag *****
on mouseDown
  do_select
end mouseDown

***** CARD #4, BUTTON #4: overhaul *****
on mouseDown
  do_select
end mouseDown

***** CARD #4, BUTTON #5: inactive *****
on mouseDown
  do_select
end mouseDown

***** CARD #5, BUTTON #1: go_next *****
on mouseUp
  go next
end mouseUp

***** CARD #6, BUTTON #1: go_next *****
on mouseUp
  go next
end mouseUp

***** CARD #7, BUTTON #1: go_next *****
on mouseUp
  go next
end mouseUp


SCRIPTS FOR STACK: templates
______________________________________________________

***** BACKGROUND #3: log model *****
-- This function gets the number of line in which thje user clicked the
-- mousebutton on
function CLICKLINE
  return trunc(((scroll of the target)
  + (item 2 of the clickloc) - (item 2 of the rect of the target
  )) div the textheight of the target) + 1
```

```
end CLICKLINE

-- This event restores the original condition of the fields after exiting
-- the card in order to make sure that the next time the log system is
-- used, the condition odf the stack is the default one
on closecard
  -- check if the field is unlocked, which means that changes have
  -- been made
  if the locktext of fld bg_fld_date is false then
    -- relock this text field
    set the locktext of fld bg_fld_date to true
  end if
  -- but if the remarks are changed, then relock all fields
  if the locktext of fld bg_fld_remarks is false then
    set cursor to 4
    lock screen
    set the locktext of fld mfgr_data to true
    set the locktext of fld pubs to true
    set the locktext of fld drawg to true
    put 0 into K_VAR
    -- make sure that at the end of each line a return is placed to
    -- avoid having text rapping arround which will mess up the line
    -- counting and missplace the date - remark coordination
    repeat until K_VAR = the number of lines of fld bg_fld_remarks
      put K_VAR + 1 into K_VAR
      put line K_VAR of fld bg_fld_remarks into CHK_LINE
      if the length of CHK_LINE > 65 then
        put return after char 65 of line K_VAR of fld bg_fld_remarks
        put K_VAR - 1 into K_VAR
      end if
    end repeat
    -- relock the text field to make menus active
    set the locktext of fld bg_fld_remarks to true
    -- restore the original screen con dition
    unlock screen
  end if
  -- exit event
end closecard

-- This event monitors the enter key in order to do a "find next" function
-- It is used because the system does not provide one, and therefore a new one
-- should be made
on enterkey
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  -- check if a search was requested
  if SEARCH_STRING_VAR is not empty then
    -- find out a global search has been requested
    if SEARCH_FIELD_VAR = "global" then
      -- ...and find the requested string
      find chars SEARCH_STRING_VAR
      -- if it was not found...
      if the result is not empty then
        -- inform the user
        answer "String not found..."
      end if
    else
      -- find out on what field the search sould be made on, and search
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      -- if not found...
      if the result is not empty then
        -- inform the user
```

```
      end if
    end if
    -- exit event
end enterkey

on newCard
  put "Find the desired card and click the button to set the transfered Repair maintenanc
e job."
  into fld click_to_set_rep_fld
end newCard

 ***** BKGND #3, FIELD #14: mfgr_data *****
on mouseDown
  global LOG_MFCTR_DATA_MENU
  get HPopupMenu (LOG_MFCTR_DATA_MENU,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 1 then
      answer "Click on the phone number you want to dial..."
      send doit to me
    end if
    if theline = 2 then
      set the locktext of me to false
    end if
  end if
end mouseDown
on doit
  wait until the mouseClick
  put the length of line CLICKLINE() of me into LAST_CHAR
  if char 1 to 5 of line CLICKLINE() of me is not "ph #:" then
    answer "Line to dial must begin with 'ph #:'"
    exit doit
  else
    put char 6 to LAST_CHAR of line CLICKLINE() of me into fld number_to_dial
    set cursor to 4
    lock screen
    show bg button tel_comp
    show bg button "phone #"
    show bg button with_modem
    show bg button with_phone
    show bg button "Pulse"
    show bg button "Tone"
    show bg button "Tone    "
    show fld number_to_dial
    show bg button close
    unlock screen
  end if
end doit

 ***** BKGND #3, FIELD #15: pubs *****
on mouseDown
  global LOG_PUBS_MENU
  get HPopupMenu (LOG_PUBS_MENU,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 1 then
    end if
    if theline = 2 then
      set the locktext of me to false
    end if
 ***** BKGND #3, FIELD #16: drawg *****
on mouseDown
  global LOG_DRWG_NUM_MENU
  get HPopupMenu (LOG_DRWG_NUM_MENU,0,the mouseV,the mouseH)

  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 1 then
    end if
    if theline = 2 then
      set the locktext of me to false
    end if
  end if
end mouseDown

 ***** BKGND #3, FIELD #17: bg_fld_date *****
on mouseDown
  global LOG_DATE_MENU
  get HPopupMenu (LOG_DATE_MENU,0,the mouseV,the mouseH)
    if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 1 then
      put the date into line CLICKLINE() of me
    end if
    if theline = 2 then
      put empty into line CLICKLINE() of me
    end if
    if theline = 3 then
      set the locktext of me to false
    end if
  end if
end mouseDown

 ***** BKGND #3, FIELD #18: bg_fld_remarks *****
on mouseDown
  global LOG_REMARK_MENU
  get HPopupMenu (LOG_REMARK_MENU,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      set the locktext of me to false
      send mouseUp to bg button go_end
    end if
    if theline = 2 then
      set the locktext of me to false
      set the scroll of me to 0
    end if
  end if
end mouseDown

 ***** BKGND #3, BUTTON #2: LightHouse *****
on mouseUp
  go to stack ARGOS
end mouseUp

 ***** BKGND #3, BUTTON #3: go_end *****
on mouseUp
  -- set the cursor to wait
  set cursor to 4
  -- lock the screen to have the fields scroll together
  set the lockscreen to true
  -- get the number of lines of the remarks field
  put the number of lines of fld bg_fld_remarks into it
  -- calculate the scroll
  put (it * 12 - 48) into it
  -- check if the scroll sould be set to zero
  if it < 0 then
    put 0 into it
  else
    -- if not...set it to the correct max scroll
    set the scroll of fld bg_fld_remarks to it
    set the scroll of fld bg_fld_date to it
  end if
  -- restore screen
  set the lockscreen to false
  -- exit event
end mouseUp

 ***** BKGND #3, BUTTON #4: go_top *****
on mouseUp
  set cursor to 4
  set the lockscreen to true
  set the scroll of fld bg_fld_remarks to 0
  set the scroll of fld bg_fld_date to 0
  set the lockscreen to false
end mouseUp

 ***** BKGND #3, BUTTON #13: down *****
on mouseUp
  put the scroll of field bg_fld_remarks into it
  add 72 to it
  set cursor to 4
  set the lockscreen to true
  set the scroll of fld bg_fld_remarks to it
  set the scroll of fld bg_fld_date to it
  set the lockscreen to false
end mouseUp

 ***** BKGND #3, BUTTON #14: up *****
on mouseUp
  put the scroll of field bg_fld_remarks into it
  subtract 72 from it
  if it < 0 then put 0 into it
  set cursor to 4
  set the lockscreen to true
  set the scroll of fld bg_fld_remarks to it
  set the scroll of fld bg_fld_date to it
  set the lockscreen to false
end mouseUp

 ***** BKGND #3, BUTTON #15: go_next *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #3, BUTTON #16: go_prev *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #3, BUTTON #17: search *****
on mouseUp
  global SEARCH_FIELD_VAR
  put "global" into SEARCH_FIELD_VAR
  lock screen
  put "Global Search" into fld search_by
  put empty into fld search_string
  show bg button search_win
  show bg button search_menu
  show bg button search_now
  show bg button forget_it
  show bg button "search for:"
  show fld search_by
  show fld search_string
  hide me
  unlock screen
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #3, BUTTON #18 *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #3, BUTTON #19: New Button *****
on mouseDown
  put "Click in the field for options" into menu1
  get HPopupMenu(menu1,0,92,50)
end mouseDown

 ***** BKGND #3, BUTTON #20: New Button *****
on mouseDown
  put "Click in the field for options" into menu1
  get HPopupMenu(menu1,0,92,200)
end mouseDown

 ***** BKGND #3, BUTTON #21: New Button *****
on mouseDown
  put "Click in the field for options" into menu1
  get HPopupMenu(menu1,0,92,350)
end mouseDown

 ***** BKGND #3, BUTTON #24: with_phone *****
on mouseUp
  set the highlight of bg button "Tone   " to true
  set the highlight of bg button "Pulse" to false
  set the highlight of bg button "Tone" to false
  dial fld number_to_dial
end mouseUp

 ***** BKGND #3, BUTTON #25: with_modem *****
on mouseUp
  if the highlight of bg button "Tone   " is true then
    answer "Pick Tone or Palse first for modem dialing..."
  else
    if the highlight of bg button "Palse" is true then
      dial empty with modem "ATH"
      dial fld number_to_dial with modem "ATDP"
    else
      dial empty with modem "ATH"
      dial fld number_to_dial with modem "ATDT"
    end if
  end if
end mouseUp

 ***** BKGND #3, BUTTON #26: Tone *****
on mouseUp
  set the highlight of me to true
```

```
  set the highlight of bg button "Tone   " to false
end mouseUp

 ***** BKGND #3, BUTTON #27: Pulse *****
on mouseUp
  set the highlight of me to true
  set the highlight of bg button "Tone" to false
  set the highlight of bg button "Tone   " to false
end mouseUp

 ***** BKGND #3, BUTTON #28: Tone    *****
on mouseUp
  set the highlight of me to true
  set the highlight of bg button "Pulse" to false
  set the highlight of bg button "Tone" to false
end mouseUp

 ***** BKGND #3, BUTTON #29: close *****
on mouseUp
  lock screen
  hide bg button tel comp
  hide bg button "phone #"
  hide bg button with_modem
  hide bg button with_phone
  hide bg button "Pulse"
  hide bg button "Tone"
  hide bg button "Tone   "
  hide fld number_to_dial
  hide me
  unlock screen
end mouseUp

 ***** BKGND #3, BUTTON #30: Print_log_rep *****
on mouseUp
  put the number of lines of fld bg_fld_remarks into LINES_COUNTED
  put 0 into TOTAL_LINES
  set cursor to 4
  lock screen
  repeat until TOTAL_LINES >= LINES_COUNTED
    put TOTAL_LINES + 43 into TOTAL_LINES
  end repeat
  put 0 into COUNT_LINES
  put 0 into GLOBAL_PAGE
  repeat until COUNT_LINES >= TOTAL_LINES
    put COUNT_LINES + 1 into PREV_COUNT_LINES
    put COUNT_LINES + 18 into COUNT_LINES
    put line PREV_COUNT_LINES to COUNT_LINES of fld bg_fld_remarka
    into REMARKS1_TO_TRANSFER
    put line PREV_COUNT_LINES to COUNT_LINES of fld bg_fld_date
    into DATE1_TO_TRANSFER
    put COUNT_LINES + 1 into PREV_COUNT_LINES
    put COUNT_LINES + 25 into COUNT_LINES
    put line PREV_COUNT_LINES to COUNT_LINES of fld bg_fld_remarka
    into REMARKS2_TO_TRANSFER
    put line PREV_COUNT_LINES to COUNT_LINES of fld bg_fld_date
    into DATE2_TO_TRANSFER
    put GLOBAL_PAGE + 1 into GLOBAL_PAGE
    push card
    go to stack Log_report
    put REMARKS1_TO_TRANSFER into fld page1_remarks_rep
    put DATE1_TO_TRANSFER into fld page1_date_rep
    go next
    put GLOBAL_PAGE into fld page_rep
    put the date into fld date_printed_rep
    open printing
    go first
    print all cards
    close printing
    pop card
  end repeat
  unlock screen
end mouseUp

 ***** BKGND #3, BUTTON #31: New Button *****
on mouseDown
  put "Click in the field for options" into menu1
  get HPopupMenu(menu1,0,142,350)
end mouseDown

 ***** BKGND #3, BUTTON #32: New Button *****
on mouseDown
  put "Click in the field for options" into menu1
  get HPopupMenu(menu1,0,142,8)
end mouseDown

 ***** BKGND #3, BUTTON #33: put_it_here *****
on mouseDown
  put return & "Put transfered Repair Job to this card" into menu
  get HPopupMenu (menu,0,125,120)
end mouseDown

on mouseUp
  global TRANSFER_ME
  global WHEN_DONE
  lock screen
  hide bg button put_it_here
  hide bg button exit_transfer_REP
  hide fld click_to_set_rep_fld
  put the number of lines of fld bg_fld_remarks into HOW_MANY_LINES
  put WHEN_DOHE into line HOW_MANY_LINES + 1 of fld bg_fld_date
  put TRANSFER_ME into line HOW_MANY_LINES + 1 of fld bg_fld_remarks
  pop card
  unlock screen
end mouseUp


 ***** BKGND #3, BUTTON #34: exit_transfer_REP *****
on mouseDown
  put "Cancel" into menu
  get HPopupMenu (menu,0,125,360)
end mouseDown

on mouseUp
  lock screen
  hide bg button put_it_here
  hide bg button exit_transfer_REP
  hide fld click_to_set_rep_fld
  pop card
  unlock screen
end mouseUp

 ***** BKGND #3, BUTTON #36: search_menu *****
on mouseDown
  global SEARCN_FIELD_VAR
  global DISPLAY
  put "Date" into menu
  put return & "Remarks" after menu
  put return & "FSN" after menu
  put return & "Unit Name" after menu
  put return & "Compartment #" after menu
  put return & "Frame #" after menu
  put return & "Side" after menu
  put return & "PN" after menu
  put return & "APL" after menu
  put return & "Inst. Date" after menu
  put return & "Spare Part Bx" after menu
  put return & "Assigned Page" after menu
  put return & "Assigned Par" after menu
  put return & "Manfr. Data" after menu
  put return & "Publications" after menu
  put return & "Drawing #" after menu
  get HPopupMenu (menu,0,the mouseV - 60,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put line theline of menu into fld search_by
    if theline = 1 then
      put "bg_fld_date" into SEARCH_FIELD_VAR
    end if
    if theline = 2 then
      put "bg_fld_remarks" into SEARCH_FIELD_VAR
    end if
    if theline = 3 then
      put "fsn" into SEARCH_FIELD_VAR
    end if
    if theline = 4 then
      put "unit" into SEARCH_FIELD_VAR
    end if
    if theline = 5 then
      put "cmpmt" into SEARCH_FIELD_VAR
    end if
    if theline = 6 then
      put "frame" into SEARCH_FIELD_VAR
    end if
    if theline = 7 then
      put "side" into SEARCH_FIELD_VAR
    end if
    if theline = 8 then
      put "pn" into SEARCH_FIELD_VAR
    end if
    if theline = 9 then
      put "apl" into SEARCH_FIELD_VAR
    end if
    if theline = 10 then
      put "inst_date" into SEARCH_FIELD_VAR
    end if
    if theline = 11 then
      put "spare_prt_bx" into SEARCH_FIELD_VAR
    end if
    if theline = 12 then
      put "pge" into SEARCH_FIELD_VAR
    end if
    if theline = 13 then
      put "patgr" into SEARCH_FIELD_VAR
    end if
    if theline = 14 then
      put "mfgr_data" into SEARCH_FIELD_VAR
    end if
    if theline = 15 then
      put "pubs" into SEARCH_FIELD_VAR
    end if
    if theline = 16 then
      put "drawg" into SEARCH_FIELD_VAR
    end if
    put line theline of menu into DISPLAY
  else
    put "Global Search" into fld search_by
    put "global" into SEARCH_FIELD_VAR
  end if
end mouseDown
on mouseUp
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #3, BUTTON #38: forget_it *****
on mouseDown
  put return & "Exit Search" into menu
  get HPopupMenu (menu,0,125,355)
end mouseDown
on mouseUp
  lock screen
  show bg button search
  hide bg button "search for:"
  hide bg button search_menu
  hide bg button search_win
  hide bg button search_now
  hide fld search_by
  hide fld search_string
  hide me
  put empty into fld search_string
  unlock screen
end mouseUp

 ***** BKGND #3, BUTTON #39: search_now *****
on mouseDown
  put return & "Find it" into menu
  get HPopupMenu (menu,0,125,335)
end mouseDown

on mouseUp
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
   set cursor to 4
  lock screen
  put fld search_string into SEARCH_STRING_VAR
  put length(SEARCH_STRING_VAR) into LAST_CHAR
  if char LAST_CHAR of SEARCH_STRING VAR = return then
    put empty into char LAST_CHAR of SEARCH_STRING_VAR
  end if
  put empty into fld search_atring
  if SEARCH_STRING_VAR is not empty then
    go next
    if SEARCH_FIELD_VAR = "global" then
      send mouseUp to bg button forget_it
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    else
      send mouseUp to bg button forget_it
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      if the result is not empty then
        answer "String not found..."
```

```
      end if
    else
      send mouseUp to bg button forget_it
    end if
    unlock screen
end mouseUp

 ***** BKGND #3, BUTTON #40: Set_me *****
-- This event adds the card to the system if it is a new card or exits
-- the log stack and returns to the driver if it is an old card
on mouseUp
  -- check whether it is a new card or a revised one
  if the short name of this stack is "log" then
    -- lock all fields
    repeat with fldno = 1 to the number of fields
      set the locktext of fld fldno to true
    end repeat
    -- if it is a revised one...
    go to card 2 of stack "PMS_driver"
  else
    -- if it is a new member...
    -- lock the screen to make things faster
    lock screen
    -- lock all fields
    repeat with fldno = 1 to the number of fields
      set the locktext of fld fldno to true
    end repeat
    -- hide del button
    hide bg button delete_me
    -- remove it from the PMS_driver
    doMenu "Cut Card"
    -- go to the log stack
    go to last card of stack log
    -- place it there
    doMenu "Paste Card"
    -- compact the stack to reduce its size
    doMenu "Compact Stack"
    --return to the PMS_driver
    go to card 2 of stack "PMS_driver"
    -- reset the screen
    unlock screen
  end if
end mouseUp

 ***** BKGND #3, BUTTON #41: delete_me *****
-- This event deletes the card and removes it from the system
on mouseUp
  -- ask the user to verify that he really wants to remove this card
  answer "Are you sure ?" with "Ok" or "Cancel"
  -- if he does...
  if it is "Ok" then
    -- lock the screen to make things ran faster
    lock screen
    -- throw the card away
    doMenu "Delete Card"
    -- compact the stack to give back to the system the space taken from this card
    doMenu "Compact Stack"
    --return to the driver
    go to card 2 of stack "PMS_driver"
    -- reset the screen
    unlock screen
  end if
  -- exit event
end mouseUp

 ***** BKGND #3, BUTTON #42: draw line *****
on mouseUp
  put "-----------------------------------------------------------------" & return after
 fld bg_fld_remarks
end mouseUp

***** BACKGROUND #4: report_graph *****
on openCard
  hide menubar
end openCard

 ***** BKGND #4, BUTTON #1: print_only_me *****
on mouseUp
  lock screen
  hide bg button print_only_me
  hide bg button lighthouse
  hide bg button go_next
  hide bg button go_prev
  hide bg button go_to_driver
  hide bg button "Performance Report"
  print this card
  show bg button print_only_me
  show bg button lighthouse
  show bg button go_next
  show bg button go_prev
  show bg button go_to_driver
  show bg button "Performance Report"
  unlock screen
end mouseUp

 ***** BKGND #4, BUTTON #2: LightHouse *****
on mouseUp
  go to card 1 of stack PMS_driver
end mouseUp

 ***** BKGND #4, BUTTON #3: go_to driver *****
on mouseUp
  go to card 2 of stack pms_driver
end mouseUp

 ***** BKGND #4, BUTTON #4: make_me *****
on mouseUp
  repeat with i = 1 to the number of lines of fld collect_info
    put item 1 of line i of fld collect_info into fld i
  end repeat
  repeat with i = 1 to the number of lines of fld collect_info
    if item 2 of line i of fld collect_info = "A" then
      put i * 5 - 4 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "B" then
      put i * 5 - 3 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "C" then
      put i * 5 - 2 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "D" then
      put i * 5 - 1 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "E" then
      put i * 5 into WHICH
      show button WHICH
    end if
  end repeat
  choose line tool
  put "86," & round(281 - (item 3 of line 1 of fld collect_info)
  * 1.5) into old_loc
  repeat with i = 2 to the number of lines of fld collect_info
    put i * 12 + 74 into h
    put round(281 - (item 3 of line i of fld collect_info) * 1.5) into v
    put h & "," & v into new_loc
    drag from old_loc to new_loc
    put new_loc into old_loc
  end repeat
  choose browse tool
  put "PMS completed :" && average_of (collect_info,3) && "%" into fld total_compl
  put the number of lines of fld collect_info into DAY_NO
  if DAY_NO = 30 then
    show button 156
  else
    if DAY_NO = 29 then
      show button 157
    else
      if DAY_NO = 28 then
        show button 158
      end if
    end if
  end if
end mouseUp

function average of fld_to,item_on
  put 0 into tot
  put 0 into count
  repeat with i = 1 to the number of lines of fld fld_to
    put tot + item item_on of line i of fld fld_to into tot
    put count + 1 into count
  end repeat
  put (tot / count) into TEMP_RES
  if TEMP_RES - trunc(TEMP_RES) > 0 then
    put (char 1 to (offset(".",TEMP_RES) + 2) of TEMP_RES) into TEMP_RES
  end if
  return (TEMP_RES)
end average_of

 ***** BKGND #4, BUTTON #5: go_next *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #4, BUTTON #6: go_prev *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #5, FIELD #7: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #5, FIELD #8: MRC *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #5, FIELD #9: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #5, FIELD #10: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #5, FIELD #11: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #5, FIELD #12: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
```

```
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #5, FIELD #13: reiman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #6, FIELD #4: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #6, FIELD #5: MRC *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #6, FIELD #6: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #6, FIELD #7: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  push card
  go to card cardName
end mouseup

 ***** BKGND #6, FIELD #8: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #6, FIELD #9: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #6, FIELD #10: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #7, FIELD #2: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #7, FIELD #3: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #7, FIELD #4: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #7, FIELD #5: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #7, FIELD #6: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #7, FIELD #7: MRC *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #7, FIELD #8: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

***** BACKGROUND #8: schedule_page1 *****
-- This function takes the contents of a container like a field and
-- sorts them acording to some item specified.  Arguments are the field
-- and the desired item to be sorted on.
-- The sorting is done in a descending order.
function fieldsort thefield,sort_on_item
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield - 1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines in thefield
      if item sort_on_item of line current of thefield > item sort_on_item of line least
linenum of thefield then
        put current into leastlinenum
      end if
    end repeat
    if leastlinenum > primary then
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsort

-- This function takes the contents of a container like a field and
-- sorts them acording to some item specified.  Arguments are the field
-- and the desired item to be sorted on.
-- The sorting is done in a ascending order.
function fieldsortnormal thefield,sort_on_item
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield - 1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines in thefield
      if item sort_on_item of line current of thefield < item sort_on_item of line least
linenum of thefield then
        put current into leastlinenum
      end if
```

```
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsortnormal

-- This function does an insertion sort of the items of a container
-- assuming that the items are allready sorted.  What it does is to pick
-- the new eilement and insert it in the correct position in the list.
-- It is used instead of a normal full sort to increase the performance
-- of the system.
function superfastdown item_to_sort,itemline
  get item_to_sort
  put line itemline of item_to_sort into PRESERVE
  delete line itemline of item_to_sort
  put item 5 of PRESERVE into TEST_ITEM
  put false into TEST
  repeat with current = itemline to the number of lines in item_to_sort
    if TEST_ITEM > item 5 of line current of item_to_sort then
      next repeat
    else
      put PRESERVE & return before line current of item_to_sort
      put true into TEST
      exit repeat
    end if
  end repeat
  if not TEST then put return & PRESERVE after last line of item_to_sort
  return item_to_sort
end superfastdown

-- This function tales a value and transforms it to screen coordinates
-- in order to plot the graph.  It takes as arguments a value and
-- returns an integer value which coresponds to the Y-axis.
function transformed_to_screen_X_coordinates item_value
  return round(160 - ( 55 * item_value / 50))
end transformed_to_screen_X_coordinates

***** BKGND #8, BUTTON #1: do_it_active *****
on mouseUp
  put the number of cards of bg schedule_page1 into NO_OF_WORK_CENTERS
  put the number of cards of bg DA_MODEL into TOTAL_DA_CARDS
  put TOTAL_DA_CARDS / NO_OF_WORK_CENTERS into DAYS_OF_QUARTER
  --delete the dublicate jobs
  push card
  put "*** START ***" & return into CHECK_JOBS_LIST
  repeat with i = 1 to NO_OF_WORK_CENTERS
    go to card i of bg schedule_page1
    repeat with LINE_COUNT = the number of lines of fld info down to 1
      put item 1 of line LINE_COUNT of fld info into CHECK_JOB
      if CHECK_JOB is in CHECK_JOBS_LIST then
        delete line LINE_COUNT of fld info
      else
        put CHECK_JOB & return after CHECK_JOBS_LIST
      end if
    end repeat
  end repeat
  put empty into CHECK_JOBS_LIST
  pop card
  put 0 into END_DATE
  repeat with i = 1 to NO_OF_WORK_CENTERS
```

```
put 0 into EVERY_2DAY_JOB_GLOBAL
put false into SORTED
put END_DATE + 1 into START_DATE
put START_DATE + DAYS_OF_QUARTER - 1 into END DATE
go to card 1 of bg schedule_page1
put fld info into JOBS_TO_BE_SCHEDULED
--set up field and var to hold mh
put empty into DAYS_STAT_MH_VAR
repeat with FILL_LINE = 1 to DAYS_OF_QUARTER
  get fld letter_sh_status of card START_DATE + FILL_LINE - 1 of
  bg DA_MODEL
  put it into STAT
  get fld bg_fld_date of card START_DATE + FILL_LINE - 1 of
  bg DA_MODEL
  put it into TEMP_DATE
  put char 1 of item 1 of TEMP_DATE into DAY_CHAR
  convert TEMP_DATE to dateitems
  put item 2 of TEMP_DATE into MONTH_NO
  put FILL_LINE & "," & DAY_CHAR & "," & MONTH NO & ","
  & STAT & "," & return into line FILL_LINE of DAYS_STAT_MH_VAR
end repeat
repeat with FILL_LINE = 0 to (DAYS_OF_QUARTER - 1)
  get fld calculated_mh of card START_DATE + FILL_LINE of bg DA_model
  put it after line (FILL_LINE + 1) of DAYS_STAT_MH_VAR
end repeat
repeat with CHANGE_S_MH = 1 to DAYS_OF_QUARTER
  if item 2 of line CHANGE_S_MH of DAYS_STAT_MH_VAR = "S" then
    put 1000 into item 5 of line CHANGE_S_MH of DAYS_STAT_MH_VAR
  end if
end repeat
--sort the var in 4321 order acording to mh (last item)
put fieldsort (DAYS_STAT_MH_VAR,5) into DAYS_STAT_MH_VAR
repeat with li = 1 to the number of lines of JOBS_TO_BE_SCHEDULED
  put 0 into ALLREADY_SCHEDULED
  put false into JOB_SCHEDULED_OK
  put char 1 of item 2 of line li of JOBS_TO_BE_SCHEDULED into TEST_CHAR
  if TEST_CHAR > "H" then
    push card  --current automatic scheduling card of new quarter stack
    if TEST_CHAR = "Q" then put 20 into DIVIDE_BY
    if TEST_CHAR = "P" then put 18 into DIVIDE_BY
    if TEST_CHAR = "O" then put 16 into DIVIDE_BY
    if TEST_CHAR = "N" then put 12 into DIVIDE_BY
    if TEST_CHAR = "M" then put 10 into DIVIDE_BY
    if TEST_CHAR = "L" then put 8 into DIVIDE_BY
    if TEST_CHAR = "K" then put 6 into DIVIDE_BY
    if TEST_CHAR = "J" then put 4 into DIVIDE_BY
    if TEST_CHAR = "I" then put 2 into DIVIDE_BY
    put item 1 of line li of JOBS_TO_BE_SCHEDULED into DEST
    if char 1 of DEST ="-" or char 1 of DEST = "." then
      delete char 1 of DEST
    end if
    go to stack PMS
    --check if card is there
    get line 3 of fld data of card DEST
    put it into RETURN_FROM_MOD_FUNCTION
    pop card
    -- check whether the job is set to every how many quarters to be scheduled
    if RETURN_FROM_MOD_FUNCTION = "MOD" then
      put item 1 of line li of JOBS_TO_BE_SCHEDULED &&
      ":  Unknown quarter desired" & return after fld not_scheduled
      next repeat
    else
      if fld cumul_quarter mod DIVIDE_BY = RETURN_FROM_MOD_FUNCTION then
        --go and schedule the job in some day of this quarter
```

```
put 0 into COUNTER
repeat until COUNTER = 10 or JOB_SCHEDULED_OK
  put COUNTER + 1 into COUNTER
  put DAYS_OF_QUARTER - ALLREADY_SCHEDULED + 1 into UP_TO_WHAT_RANDOM
  put random(UP_TO_WHAT_RANDOM) into RANDOMLY_PICKED_DAY
  if item 4 of line RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR is in
  item 4 of line li of JOBS_TO_BE_SCHEDULED then
    get the short name of card ((item 1 of line
    RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR)
    + START_DATE - 1) of bg DA_MODEL
    put it into WHEN_TO_SCHEDULE
    repeat with WC_involved = 1 to the number of words of item 6
      of line li of JOBS_TO_BE_SCHEDULED
      put word WC_INVOLVED of item 6 of line li of JOBS_TO_BE_SCHEDULED
      into WHAT_WC
      put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
      delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
      put WHAT_WC before WHEN_TO_SCHEDULE
      put item 1 of line li of JOBS_TO_BE_SCHEDULED & return
      after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
      put item 3 of line li of JOBS_TO_BE_SCHEDULED & return
      aftet fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
      add item 3 of line li of JOBS_TO_BE_SCHEDULED to fld
      calculated_mh of card WHEN_TO_SCHEDULE
    end repeat
    add item 3 of line li of JOBS_TO_BE_SCHEDULED to
    item 5 of line RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR
    put TRUE into JOB_SCHEDULED_OK
    put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
    put superfastdown(DAYS_STAT_MH_VAR,RANDOMLY_PICKED_DAY) into DAYS_STAT_MH
_VAR
  end if
end repeat
if not JOB_SCHEDULED_OK then
  put 0 into CHECK_DAY
  tepeat until  CHECK_DAY =  DAYS_OF_QUARTER or JOB_SCHEDULED_OK
    put CHECK_DAY + 1 into CHECK_DAY
 if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
    item 4 of line li of JOBS_TO_BE_SCHEDULED then
      get the short name of card ((item 1 of line
      CHECK_DAY of DAYS_STAT_MH_VAR)
      + START_DATE - 1) of bg DA_MODEL
      put it into WHEN_TO_SCHEDULE
      repeat with WC_involved = 1 to the number of words of item 6
        of line li of JOBS_TO_BE_SCHEDULED
        put word WC_INVOLVED of item 6 of line li of JOBS_TO_BE_SCHEDULED
        into WHAT_WC
        put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
        delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
        put WHAT_WC before WHEN_TO_SCHEDULE
        put item 1 of line li of JOBS_TO_BE_SCHEDULED & return
        after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
        put item 3 of line li of JOBS_TO_BE_SCHEDULED & return
        after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
        add item 3 of line li of JOBS_TO_BE_SCHEDULED to fld
        calculated_mh of card WHEN_TO_SCHEDULE
      end repeat
      add item 3 of line li of JOBS_TO_BE_SCHEDULED to
      item 5 of line CHECK_DAY of DAYS_STAT_MH_VAR
      put TRUE into JOB_SCHEDULED_OK
      put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
      put superfastdown(DAYS_STAT_MH_VAR,CHECK_DAY) into DAYS_STAT_MH_VAR
    end if
  end repeat
```

```
          end if
          if not JOB_SCHEDULED_OK then
            put item 1 of line 11 of JOBS_TO_BE_SCHEDULED &&
            ":  No appropriate day found" & return after fld not_scheduled
          end if

        end if --cumul quarter check
      end if --return from mod function
    end if --if check char is  > than H
    if TEST_CHAR = "H" then --quarterly scheduled job
  --go and schedule the job in some day of this quarter
      put D into COUNTER
      repeat until COUNTER = 10 or JOB_SCHEDULED_OK
        put COUNTER + 1 into COUNTER
        put DAYS_OF_QUARTER - ALLREADY_SCHEDULED + 1 into UP_TO_WHAT_RANDOM
        put random(UP_TO_WHAT_RANDOM) into RANDOMLY_PICKED_DAY
        if item 4 of line RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR is in
        item 4 of line 11 of JOBS_TO_BE_SCHEDULED then
          get the short name of card ((item 1 of line
          RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR)
          + START_OATE - 1) of bg DA_MODEL
          put it into WHEN_TO_SCHEDULE
          repeat with WC_involved = 1 to the number of words of item 6
            of line 11 of JOBS_TO_BE_SCHEDULED
            put word WC_INVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
            into WHAT_WC
            put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
            delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
            put WHAT_WC before WHEN_TO_SCHEDULE
            put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
            after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
            put item 3 of line 11 of JOBS_TO_BE_SCHEDULED & return
            after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
            add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
            calculated_mh of card WHEN_TO_SCHEDULE
          end repeat
          add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to
          item 5 of line RANDOMLY_PICKED_DAY of DAYS_STAT_MH_VAR
          put TRUE into JOB_SCHEDULED_OK
          put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
          put superfastdown(DAYS_STAT_MH_VAR,RANDOMLY_PICKED_DAY) into DAYS_STAT_MH_VA
R
        end if
      end repeat
      if not JOB_SCNEDULED_OK then
        put D into CHECK_DAY
        repeat until CHECK_DAY = DAYS_OF_QUARTER or JOB_SCHEDULED_OK
          put CHECK_DAY + 1 into CHECK_DAY
          if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
          item 4 of line 11 of JOBS_TO_BE_SCHEDULED then
            get the short name of card ((item 1 of line
            CHECK_DAY of DAYS_STAT_MH_VAR)
            + START_DATE - 1) of bg DA_MODEL
            put it into WHEN_TO_SCHEDULE
            repeat with WC_involved = 1 to the number of words of item 6
              of line 11 of JDBS_TO_BE_SCHEDULED
              put word WC_INVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
              into WHAT_WC
              put offset("-",WHEN_TO_SCHEDULE) into DFFSET_COUNT
              delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
              put WNAT_WC before WHEN_TO_SCNEDULE
              put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
              after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
              after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
              add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
              calculated_mh of card WHEN_TO_SCHEDULE
            end repeat
            add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to
            item 5 of line CHECK_DAY of DAYS_STAT_MH_VAR
            put TRUE into JOB_SCHEDULED_OK
            put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
            put superfastdown(DAYS_STAT_MH_VAR,CHECK_DAY) into DAYS_STAT_MH_VAR
          end if
        end repeat
      end if
      if not JOB_SCHEDULED_OK then
        put item 1 of line 11 of JOBS_TO_BE_SCHEDULED &&
        ":  No appropriate day found" & return after fld not_scheduled
      end if
    end if --if check char is = N
    if TEST_CHAR = "G" then --2M scheduled job
      push card
      put item 1 of line 11 of JOBS_TO_BE_SCHEDULED into DEST
      if char 1 of DEST ="-" or char 1 of DEST = "." then
        delete char 1 of DEST
      end if
      go to stack PMS
      --check if card is there
      get line 3 of fld data of card DEST
      put it into RETURN_FROM_MOD_FUNCTION
      pop card
      -- check whether the job is set to every how many quarters to be scheduled
      if RETURN_FROM_MOD_FUNCTION = "MOD" then
        put item 1 of line 11 of JOBS_TO_BE_SCHEDULED &&
        ":  Unknown month desired" & return after fld not_scheduled
        next repeat
      else
        put D into CHECK_DAY
        put D into ALL_MONTHS
        put "Months:" into SCHEDULED_MONTH
        repeat until  CHECK_DAY =  DAYS_DF_QUARTER or ALL_MONTHS = 2
          put CHECK_DAY + 1 into CHECK_DAY
          if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
          item 4 of line 11 of JOBS_TO_BE_SCHEDULED and
          item 3 of line CHECK_DAY of DAYS_STAT_MH_VAR is not in
          SCHEDULED_MONTH then
            get the short name of card ((item 1 of line
            CHECK_DAY of DAYS_STAT_MN_VAR)
            + START_DATE - 1) of bg DA_MODEL
            put it into WHEN_TO_SCHEDULE
            repeat with WC_involved = 1 to the number of words of item 6
              of line 11 of JOBS_TO_BE_SCNEDULED
              put word WC_INVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
              into WHAT_WC
              put offset("-",WHEN_TO_SCNEDULE) into OFFSET_COUNT
              delete char 1 to (OFFSET_COUNT - 1) of WNEN_TD_SCHEDULE
              put WHAT_WC before WHEN_TO_SCHEDULE
              put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
              after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
              put item 3 of line 11 of JOBS_TD_BE_SCHEDULED & return
              after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
              add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
              calculated_mh of card WNEN_TO_SCHEDULE
            end repeat
            add item 3 of line 11 of JOBS_TO_BE_SCNEDULED to
            item 5 of line CHECK_DAY of DAYS_STAT_MH_VAR
            put item 3 of line CHECK_DAY of DAYS_STAT_MH_VAR & "," after
            SCHEDULED_MONTH
            put ALL_MONTHS + 1 into ALL_MONTHS
            put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
            put superfastdown(DAYS_STAT_MH_VAR,CHECK_DAY) into DAYS_STAT_MH_VAR
          end if
        end repeat
      end if
    end if --if check char is = G
    if TEST_CHAR = "F" then --Monthly scheduled job
      put 0 into CHECK_DAY
      put 0 into ALL_MONTHS
      put "Months:" into SCHEDULED_MONTH
      repeat until  CHECK_DAY =  DAYS_OF_QUARTER or ALL_MONTHS = 3
        put CHECK_DAY + 1 into CHECK_DAY
        if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
        item 4 of line ii of JOBS_TO_BE_SCHEDULED and
        item 3 of line CHECK_DAY of DAYS_STAT_MH_VAR is not in
        SCHEDULED_MONTH then
          get the short name of card ((item 1 of line
          CHECK_DAY of DAYS_STAT_MH_VAR)
          + START_DATE - 1) of bg DA_MODEL
          put it into WHEH_TO_SCHEDULE
          repeat with WC_involved = 1 to the number of words of item 6
            of line ii of JOBS_TO_BE_SCHEDULED
            put word WC_IHVOLVED of item 6 of line ii of JOBS_TO_BE_SCHEDULED
            into WHAT_WC
            put offset("-",WHEH_TO_SCHEDULE) into OFFSET_COUNT
            delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
            put WHAT_WC before WHEN_TO_SCHEDULE
            put item 1 of line ii of JOBS_TO_BE_SCHEDULED & return
            after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
            put item 3 of line ii of JOBS_TO_BE_SCHEDULED & return
            after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
            add item 3 of line ii of JOBS_TO_BE_SCHEDULED to fld
            calculated_mh of card WHEN_TO_SCHEDULE
          end repeat
          add item 3 of line ii of JOBS_TO_BE_SCHEDULED to
          item 5 of line CHECK_DAY of DAYS_STAT_MH_VAR
          put TRUE into JOB_SCHEDULED_OK
          put item 3 of line CHECK_DAY of DAYS_STAT_MH_VAR & "," after
          SCHEDULED_MONTH
          put ALL_MONTHS + 1 into ALL_MONTHS
          put ALLREADY_SCHEDULED + 1 into ALLREADY_SCHEDULED
          put superfastdown(DAYS_STAT_MH_VAR,CHECK_DAY) into DAYS_STAT_MH_VAR
        end if
      end repeat
    end if --if check char is = F
    if TEST_CNAR = "E" then -- 2-Week scheduled job
      if not SORTED then
        put fieldsortnormal (DAYS_STAT_MH_VAR,1) into DAYS_STAT_MH_VAR
        put true into SORTED
      end if
      put 0 into WHAT_2WEEK_BLOCK
      put 0 into TO_DATE
      repeat for DAYS_OF_QUARTER div 14
        put WHAT_2WEEK_BLOCK + 1 into WHAT_2WEEK_BLOCK
        put TO_DATE + 1 into FROM_DATE
        put FROM_DATE + 13 into TO_DATE
        put line FROM_DATE to TO_DATE of DAYS_STAT_MN_VAR into
        TEMP_DAYS_STAT_MH_VAR
        put fieldsortnormal (TEMP_DAYS_STAT_MH_VAR,5) into TEMP_DAYS_STAT_MH_VAR
        repeat with CHECK_DAY = 1 to 14
          if item 4 of line CHECK_DAY of TEMP_DAYS_STAT_MH_VAR is in
          item 4 of line ii of JOBS_TO_BE_SCHEDULED then
            get the short name of card ((item 1 of line
            CHECK_DAY of TEMP_DAYS_STAT_MH_VAR)
            + START_DATE - 1) of bg DA_MODEL
            put it into WHEN_TO_SCHEDULE
            repeat with WC_involved = 1 to the number of words of item 6
              of line ii of JOBS_TO_BE_SCHEDULED
              put word WC_IHVOLVED of item 6 of line ii of JOBS_TO_BE_SCHEDULED
              into WHAT_WC
              put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
              delete char 1 to (OFFSET_COUHT - 1) of WHEN_TO_SCNEDULE
              put WHAT WC before WHEN_TO_SCHEDULE
              put item 1 of line ii of JOBS_TO_BE_SCHEDULED & return
              after fld bg_fld_MRC_no of card WHEH_TO_SCHEDULE of bg DA_model
              put item 3 of line ii of JOBS_TO_BE_SCHEDULED & return
              after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
              add item 3 of line ii of JOBS_TO_BE_SCHEDULED to fld
              calculated_mh of card WHEN_TO_SCHEDULE
            end repeat
            add item 3 of line ii of JOBS_TO_BE_SCHEDULED to
            item 5 of line (item 1 of line CHECK_DAY of
            TEMP_DAYS_STAT_MH_VAR) of DAYS_STAT_MH_VAR
            put TRUE into JOB_SCHEDULED_OK
            exit repeat
          end if
        end repeat
        --report that job was not scheduled on this 15day block
        if not JOB_SCHEDULED_OK then
          put item 1 of line ii of JOBS_TO_BE_SCHEDULED &&
          ":  unscheduled in 2W block" && WHAT_2WEEK_BLOCK
          & return after fld not_scheduled
        end if
      end repeat
    end if --if check char is = E
    if TEST_CHAR = "D" then -- Weekly scheduled job
      if not SORTED then
        put fieldsortnormal (DAYS_STAT_MH_VAR,1) into DAYS_STAT_MH_VAR
        put true into SORTED
      end if
      put 0 into WHAT_WEEK_BLOCK
      put 0 into TO_DATE
      repeat for DAYS_DF_QUARTER div 7
        put WHAT_WEEK_BLOCK + 1 into WHAT_WEEK_BLOCK
        put TO_DATE + 1 into FROM_DATE
        put FROM_DATE + 6 into TO_DATE
        put line FROM_DATE to TD_DATE of DAYS_STAT_MH_VAR into
        TEMP_DAYS_STAT_MH_VAR
        put fieldsortnormal (TEMP_DAYS_STAT_MH_VAR,5) into TEMP_DAYS_STAT_MH_VAR
        repeat with CHECK_DAY = 1 to 7
          if item 4 of line CHECK_DAY of TEMP_DAYS_STAT_MH_VAR is in
          item 4 of line ii of JOBS_TO_BE_SCHEDULED then
            get the short name of card ((item 1 of line
            CHECK_DAY of TEMP_DAYS_STAT_MH_VAR)
            + START_DATE - 1) of bg DA_MODEL
            put it into WHEH_TO_SCHEDULE
            repeat with WC_involved = 1 to the number of words of item 6
              of line ii of JOBS_TO_BE_SCHEDULED
              put word WC_INVOLVED of item 6 of line ii of JOBS_TO_BE_SCHEDULED
              into WHAT_WC
              put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
              delete char 1 to (OFFSET_COUNT - 1) of WHEH_TO_SCHEDULE
              put WHAT_WC before WHEN_TO_SCHEDULE
              put item 1 of line ii of JOBS_TO_BE_SCNEDULED & return
              after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
            put item 3 of line 11 of JOBS_TO_BE_SCHEDULED & return
            after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
            add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
            calculated_mh of card WHEH_TO_SCHEDULE
          end repeat
          add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to
          item 5 of line (item 1 of line CHECK_DAY of
          TEMP_DAYS_STAT_MH_VAR) of DAYS_STAT_MH_VAR
          put TRUE into JOB_SCHEDULED_OK
          exit repeat
        end if
      end repeat
      --report that job was not scheduled on this 15day block
      if not JOB_SCHEDULED_OK then
        put item 1 of line 11 of JOBS_TO_BE_SCHEDULED &&
        ":  unscheduled in 7day block" && WHAT_WEEK_BLOCK
        & return after fld not_scheduled
      end if
    end repeat
  end if --if check char is = D
  if TEST_CHAR = "C" then -- 3day scheduled job
    if not SORTED then
      put fieldsortnormal (DAYS_STAT_MH_VAR,1) into DAYS_STAT_MH_VAR
      put true into SORTED
    end if
    put EVERY_3DAY_JOB_GLOBAL + 1 into EVERY_3DAY_JOB_GLOBAL
    put (EVERY_3DAY_JOB_GLOBAL mod 3) + 1 into THE_DATE
    repeat while  THE_DATE <=  DAYS_OF_QUARTER
      if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
      item 4 of line 11 of JOBS_TO_BE_SCHEDULED then
        get the short name of card ((item 1 of line
        THE_DATE of DAYS_STAT_MH_VAR)
        + START_DATE - 1) of bg DA_MODEL
        put it into WHEH_TO_SCHEDULE
        repeat with WC_involved = 1 to the number of words of item 6
          of line 11 of JOBS_TO_BE_SCHEDULED
          put word WC_IHVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
          into WHAT_WC
          put offset("-",WHEN_TO_SCHEDULE) into OFFSET_COUNT
          delete char 1 to (OFFSET_COUHT - 1) of WHEN_TO_SCHEDULE
          put WHAT_WC before WHEH_TO_SCHEDULE
          put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_MRC_no of card WHEH_TO_SCHEDULE of bg DA_model
          put item 3 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_tot_mh of card WHEH_TO_SCHEDULE
          add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
          calculated_mh of card WHEH_TO_SCHEDULE
        end repeat
      end if
      put THE_DATE + 3 into THE_DATE
    end repeat
  end if --if check char is = C
  if TEST_CHAR = "B" then -- 2day scheduled job
    if not SORTED then
      put fieldsortnormal (DAYS_STAT_MH_VAR,1) into DAYS_STAT_MH_VAR
      put true into SORTED
    end if
    put EVERY_2DAY_JOB_GLOBAL + 1 into EVERY_2DAY_JOB_GLOBAL
    put (EVERY_2DAY_JOB_GLOBAL mod 2) + 1 into THE_DATE
    repeat while  THE_DATE <=  DAYS_OF_QUARTER
      if item 4 of line CHECK_DAY of DAYS_STAT_MH_VAR is in
      item 4 of line 11 of JOBS_TO_BE_SCHEDULED then
        get the short name of card ((item 1 of line
```

```
        + START_DATE - 1) of bg DA_MODEL
        put it into WHEH_TO_SCHEDULE
        repeat with WC_involved = 1 to the number of words of item 6
          of line 11 of JOBS_TO_BE_SCHEDULED
          put word WC_IHVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
          into WHAT_WC
          put offset("-",WHEH_TO_SCHEDULE) into OFFSET_COUNT
          delete char 1 to (OFFSET_COUNT - 1) of WHEH_TO_SCHEDULE
          put WHAT_WC before WHEN_TO_SCHEDULE
          put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
          put item 3 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_tot_mh of card WHEH_TO_SCHEDULE
          add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
          calculated_mh of card WHEN_TO_SCHEDULE
        end repeat
      end if
      put THE_DATE + 2 into THE_DATE
    end repeat
  end if --if check char is = B
  if TEST_CHAR = "A" then -- Every day scheduled job
    if not SORTED then
      put fieldsortnormal (DAYS_STAT_MH_VAR,1) into DAYS_STAT_MH_VAR
      put true into SORTED
    end if
    put 1 into THE_DATE
    repeat while  THE_DATE <=  DAYS_OF_QUARTER
      if item 4 of line THE_DATE of DAYS_STAT_MH_VAR is in
      item 4 of line 11 of JOBS_TO_BE_SCHEDULED then
        get the short name of card ((item 1 of line
        THE_DATE of DAYS_STAT_MH_VAR)
        + START_DATE - 1) of bg DA_MODEL
        put it into WHEN_TO_SCHEDULE
        repeat with WC_involved = 1 to the number of words of item 6
          of line 11 of JOBS_TO_BE_SCHEDULED
          put word WC_IHVOLVED of item 6 of line 11 of JOBS_TO_BE_SCHEDULED
          into WHAT_WC
          put offset("-",WHEH_TO_SCHEDULE) into OFFSET_COUNT
          delete char 1 to (OFFSET_COUNT - 1) of WHEN_TO_SCHEDULE
          put WHAT_WC before WHEN_TO_SCHEDULE
          put item 1 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_MRC_no of card WHEN_TO_SCHEDULE of bg DA_model
          put item 3 of line 11 of JOBS_TO_BE_SCHEDULED & return
          after fld bg_fld_tot_mh of card WHEN_TO_SCHEDULE
          add item 3 of line 11 of JOBS_TO_BE_SCHEDULED to fld
          calculated_mh of card WHEN_TO_SCHEDULE
        end repeat
      end if
      put THE_DATE + 1 into THE_DATE
    end repeat
  end if --if check char is = A
end repeat --for number of jobs
-- calculate total mh per day
repeat with DAYS_MH_COUHT = START_DATE to END_DATE
  get fld bg_fld_date of card DAYS_MH_COUNT of bg DA_model
  convert it to abbreviated date
  put it && ": " after card fld days_mh
  get fld calculated_mh of card DAYS_MH_COUNT of bg DA_model
  put it & return after card fld days_mh
end repeat
--draw graphs
put empty into GRAPH_VAR
repeat with LINE_COUNT = 1 to the number of lines of card fld days_mh
```

```
      days_mh & return after GRAPH_VAR
    end repeat
    repeat with LINE_COUNT = 1 to the number of lines of GRAPH_VAR
      put transformed_to_screen_X_coordinates (line LINE_COUNT
      of GRAPH_VAR) into line LINE_COUNT of GRAPH_VAR
    end repeat
    choose line tool
    put 264 into PIXEL_COUNT
    put PIXEL_COUNT & "," & line 1 of GRAPH_VAR into OLD_LOC
    repeat with DAY_COUNT = 2 to the number of lines of GRAPH_VAR
      put PIXEL_COUNT + 2 into PIXEL_COUNT
      put PIXEL_COUNT & "," & line DAY_COUNT of GRAPH_VAR into NEW_LOC
      drag from OLD_LOC to NEW_LOC
      put NEW_LOC into OLD_LOC
    end repeat
    choose browse tool
  end repeat --for number of work centers
  doMenu "Compact Stack"
end mouseUp

 ***** BKGND #8, BUTTON #3: print_this_fld *****
on mouseDown
  global TRANSFERED_DATA
  put "MIP #" into popup
  put return & "MRC #" after popup
  put return & "MH destibution" after popup
  put return & "Scheduling information" after popup
  put return & "Unscheduled jobs" after popup
  get HPopupMenu(popup,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      put "fld WC_mips" into print_what
      put "MIP # FOR WC:" && fld WC_name & return into temp_label
    end if
    if theline = 2 then
      put "fld WC_mrcs" into print_what
      put "MRC # FOR WC:" && fld WC_name & return into temp_label
    end if
    if theline = 3 then
      put "card fld days_mh" into print_what
      put "DAYS/MH DESTRIBUTION FOR WC:" && fld WC_name & return into temp_label
    end if
    if theline = 4 then
      put "fld not_scheduled" into print_what
      put "MIP/MRC JOBS NOT SCHEDULED FOR WC:" && fld WC_name & return into temp_label
    end if
    if theline = 5 then
      put "fld info" into print_what
      put "SCHEDULING INFORMATION FOR WC:" && fld WC_name & return
      into temp_label
    end if
    put "PRINTED ON:" &"     " & the date &"     " & the time & return
    & return into TRANSFERED_DATA
    put "WORK CENTER :" && fld WC_name & return &
    "QUARTER :" && fld quarter_to_schedule & return &
    "QUARTER AFTER OVHL :" && fld Q_after_ovhl & return &
    "SHIP STATUS :" && fld sh_status & return & return
    after TRANSFERED_DATA
    put temp_label & return & return after TRANSFERED_DATA
    put the value of print_what after TRANSFERED_DATA
    set cursor to 4
    push card
    lock screen
    go to stack generic_fld_print
    send mouseUp to bg button print_me
    pop card
    unlock screen
    put empty into TRANSFERED_DATA
  end if
end mouseDown

***** BACKGROUND #9: DA_model *****
function CLICKLINE
  return trunc(((scroll of the target)
  + (item 2 of the clickloc) - (item 2 of the rect of the target
  )) div the textheight of the target) + 1
end CLICKLINE

function NEWCLICKLINE
  return trunc(((scroll of fld bg_fld_job_compl_by)
  + (item 2 of the clickloc) - (item 2 of the rect of fld bg_fld_job_compl_by
  )) div the textheight of fld bg_fld_job_compl_by) + 1
end NEWCLICKLINE

on openCard
  hide menuBar
  if fld bg_fld_date = the long date and the highlight of button
  rescheduled is false then
    reschedule
  end if
end openCard

on reschedule
  show bg button "Rescheduling..."
  lock screen
  put fld bg_fld_date into TEMP_DATE
  put "X" into fld bg_fld_date
  put the id of this card into TEMP_ID
  put fld bg_fld_wc_DA into it
  repeat until fld bg_fld_wc_DA <> it or the highlight of button
    rescheduled is true
    go prev
  end repeat
  if fld bg_fld_wc_DA <> it then
    go next
  end if
  repeat until the id of this card = TEMP_ID
    put empty into MRC_NO_TEMP
    put empty into DAYS_TEMP
    put empty into REP_TEMP
    put empty into REP_DAYS_TEMP
    put 0 into MH_TEMP
    repeat with i = 1 to the number of lines of fld bg_fld_MRC_no
      if line i of fld bg_fld_PMS_check is empty and
      line i of fld bg_fld_MRC_no is not empty then
        put line i of fld bg_fld_MRC_no & return after MRC_NO_TEMP
        put line i of fld bg_fld_PMS_days_fwd + 1 & return after DAYS_TEMP
        put line i of fld bg_fld_tot_mh & return after MH_TEMP
      end if
    end repeat
    repeat with i = 1 to the number of lines of fld bg_fld_rep_maint
      if line i of fld bg_fld_REP_check is empty and
      line i of fld bg_fld_rep_maint is not empty then
        put line i of fld bg_fld_rep_maint & return after REP_TEMP
        put line i of fld bg_fld_REP_days_fwd + 1 & return after REP_DAYS_TEMP
      end if
    end repeat
    go next
    put MRC_NO_TEMP after fld bg_fld_MRC_no
    put DAYS_TEMP after fld bg_fld_PMS_days_fwd
    put MH_TEMP after fld bg_fld_tot_mh
    put 0 into ADD_TEMP
    repeat with jobs = 1 to the number of lines of fld bg_fld_tot_mh
      put ADD_TEMP + line jobs of fld bg_fld_tot_mh into ADD_TEMP
    end repeat
    put ADD_TEMP into fld calculated_mh
    set the highlight of button rescheduled to true
  end repeat
  put TEMP_DATE into fld bg_fld_date
  hide bg button "Rescheduling..."
  unlock screen
end reschedule

on activate_Top
  global TARGETLINE
  global NOW_MANY_FIELDS
  global TEMPITEM
  global TEMPLINE
  global GLOBAL_JOB
  global DA_PMS_MENU
  put empty into NOW_MANY_FIELDS
  get HPopupMenu(DA_PMS_MENU,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 1 and theItem = 2 then
      check_job
    end if
    if theline = 1 and theItem = 3 then
      uncheck_job
    end if
    if theline = 1 and theItem = 4 then
      cancel_job
    end if
    if theline = 2 and theItem = 2 then
      new_note
    end if
     if theline = 2 and theitem = 3 then
      see_note
    end if
    if theline = 2 and theitem = 4 then
      see_all_notes
    end if
    if theline = 3 and theItem = 2 then
      lock screen
      push card
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to card DESTINATION of stack PMS
      put line 2 of fld data into it
      go to card it of stack log
      unlock screen
    end if
     if theline = 3 and theItem = 3 then
      if line TARGETLINE of fld bg_fld_PMS_CNECK = "\326" then
        lock screen
        push card
        put line TARGETLINE of fld bg_fld_MRC_no && "," && line TARGETLINE
        of fld bg_fld_comments into GLOBAL_JOB
        put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
        put fld bg_fld_data into WHEN
        go to card DESTINATION of stack PMS
        put line 2 of fld data into it
        go to card it of stack log
        put the number of lines of fld bg_fld_remarks into NOW_MANY_LINES
        put WHEN into line HOW_MANY_LINES + 1 of fld bg_fld_date
        put GLOBAL_JOB into line HOW_MANY_LINES + 1 of fld bg_fld_remarks
        unlock screen
      else
        answer "Job must be completed to be transfered."
      end if
    end if
    if theline = 4 and theitem = 2 then
      set cursor to 4
      push card
      lock screen
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to stack PMS
      go to card (item 2 of line binary(mrc_list,DESTINATION) of fld
      mrc_list)
      unlock screen
    end if
    if theline = 4 and theitem = 3 then
      set cursor to 4
      push card
      lock screen
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to stack PMS
      go to card (item 2 of line binary(mrc_list,DESTINATION) of fld
      mrc_list)
      put line 1 of fld data into TEMPMIP
      delete char 1 to 4 of TEMPMIP
      go first
      go to card (item 2 of line binary(mip_list,TEMPMIP) of fld
      mip_list)
      unlock screen
    end if
    if theline = 5 then
      put theItem into TEMPITEM
      put theline into TEMPLINE
      add_short_comment
    end if
    if theline = 6 and theitem = 2 then
      -- set cursor to wait
      set cursor to 4
      -- lock screen to let things roll faster
      lock screen
      -- get WC to look for
      put fld bg_fld_wc_DA into TEMP_WC
      -- preserve this card
      push card
      -- find first card of personel bg
      go to card 1 of bg personel_bg of stack ship_data
      -- find the desired WC
      find TEMP_WC in fld WC_personel
      -- get the names of the personel
      put fld ship_data_name into TEMP_NAMES
      -- go back
      pop card
      -- put the names to personel fld
      put TEMP_NAMES into fld bg_fld_avail_personel
      -- show the personel names
      show fld bg_fld_avail_personel
      - show close button
      show bg button bg_but_close_personel
      -- reactivate the screen
      unlock screen
    end if
    if theline = 6 and theItem = 3 then
      -- delete the person assigned
      put empty into line CLICKLINE() of fld bg_fld_job_compl_by
    end if
  end if --not zero
end activate_Top

on activate_Bottom
  global TARGETLINE
  global DA_REP_MAINT_MENU
  get HPopupMenu(DA_REP_MAINT_MENU,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theItem
    if theline = 1 and theItem = 2 then
      add_repair_job
    end if
    if theline = 1 and theItem = 3 then
      remove_repair_job
    end if
    if theline = 2 and theItem = 2 then
      check_repair_job
    end if
    if theline = 2 and theItem = 3 then
      uncheck_repair_job
    end if
    if theline = 2 and theItem = 4 then
      cancel_repair_job
    end if
    if theline = 3 and theItem = 2 then
      fill_note_repair_job
    end if
    if theline = 3 and theItem = 3 then
      see_note_repair_job
    end if
    if theline = 3 and theItem = 4 then
      see_all_notes
    end if
    if theline = 4 and theItem = 2 then
      go_REP_to_log
    end if
    if theline = 4 and theItem = 3 then
      transfer_REP_to_log
    end if
  end if
end activate_Bottom

on check_job
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_MRC_no is not empty then
    put "\326" into line TARGETLINE of fld bg_fld_PMS_check
    put 0 into PMS_COMPLETED
    put 0 into JOBS_TO_DO
    repeat with i = 1 to the number of lines of fld bg_fld_MRC_no
      put JOBS_TO_DO + 1 into JOBS_TO_DO
      if line i of fld bg_fld_PMS_check = "\326" then
        put PMS_COMPLETED + 1 into PMS_COMPLETED
      end if
    end repeat
    put the trunc of (PMS_COMPLETED * 100/JOBS_TO_DO) into fld completed
  end if
end check_job

on uncheck_job
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_MRC_no is not empty then
    put empty into line TARGETLINE of fld bg_fld_PMS_check
    put empty into line TARGETLINE of fld bg_fld_comments
    put 0 into PMS_COMPLETED
    put 0 into JOBS_TO_DO
    repeat with i = 1 to the number of lines of fld bg_fld_MRC_no
      put JOBS_TO_DO + 1 into JOBS_TO_DO
      if line i of fld bg_fld_PMS_check = "\326" then
        put PMS_COMPLETED + 1 into PMS_COMPLETED
      end if
    end repeat
    put the trunc of (PMS_COMPLETED * 100/JOBS_TO_DO) into fld completed
  end if
end uncheck_job

on cancel_job
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_MRC_no is not empty then
    if line TARGETLINE of fld bg_fld_PMS_check is not "\326" then
      put "-" into line TARGETLINE of fld bg_fld_PMS_check
    else
      answer "You may not cancel a completed job."
    end if
  end if
end cancel_job

on new_note
  global TARGETLINE
  global HOW_MANY_FIELD
  lock screen
  if line TARGETLINE of fld bg_fld_note_id is empty then
    --make new fld
    put the number of card fields into HOW_MANY_FIELDS --chk # of old flds
    put HOW_MANY_FIELDS + 1 into HOW_MANY_FIELDS --add one for new fld
    choose field tool
    drag from 242,184 to 506,314 with commandkey
    --set new field properties
    set the style of card field HOW_MANY_FIELDS to scrolling --set style
    set the textsize of card fld HOW_MANY_FIELDS to 9 --set font size
    set the textfont of card fld HOW_MANY_FIELDS to Geneva --set font
  else
    set the locktext of card field HOW_MANY_FIELDS to false
    show card field HOW_MANY_FIELDS
  end if
  --show new buttons
  show bg button bg_but_frame
  show bg button bg_but_print_note
  show bg button bg_but_del_note
  show bg button bg_but_close_note
  --fill in info
  put line TARGETLINE of fld bg_fld_MRC_no
  && "                 " && the short date && "          "
  && the time & return after card fld HOW_MANY_FIELDS
  put "................................................................................"
  & return & return after card fld HOW_MANY_FIELDS
  --return to normal
  choose browse tool
  unlock screen
```

```
end new_note

on see_note
  global TARGETLINE
  --connect to activator
  put line TARGETLINE of fld bg_fld_note_id into it
  if it is not empty then
    show card fld id it
    --show buttons
    show bg button bg_but_frame
    show bg button bg_but_print_note
    show bg button bg_but_del_note
    show bg button bg_but_close_note
  end if
end see_note

on see_all_notes
  lock screen
  repeat with i=1 to the number of card fields
    if the visible of card fld i is true then
      answer "You mast close the note first..."
      exit see_all_notes
    end if
  end repeat
  hide bg button bg_but_frame
  hide bg button bg_but_print_note
  hide bg button bg_but_del_note
  hide bg button bg_but_close_note
  put empty into fld bg_fld_all_notes
  repeat with i=1 to the number of card fields
    put card fld i after fld bg_fld_all_notes
    repeat with ii=1 to 3
      put return after fld bg_fld_all_notes
    end repeat
  end repeat
  show fld bg_fld_all_notes
  --show buttons
  show bg button bg_but_frame_all_notes
  show bg button bg_but_print_note_all
  show bg button bg_but_close_note_all
  unlock screen
end see_all_notes

on add_short_comment
  global templtem
  global TEMRLINE
  global DA_PMS_MENU
  global TARGETLINE
  get item templtem of line TEMPLINE of DA_PMS_MENU
  if it is "OTHER" then
    set cursor to 4
    ask "Type in a SNORT comment...(30 char max)"
    put it into STRTOCHECK
    put empty into STRSTORAGE
    put empty into FINALSTR
    if length (STRTOCNECK) > 30 then
      answer "String too long...it will be trancated."
      with "CANCEL" or "OK"
      if it is "OK" then
        put empty into NEWSTR
        repeat with i = 1 to 30
          put char i of STRTOCNECK after NEWSTR
        end repeat
      else
        exit add_short_comment
      end if
    else
      put STRTOCHECK into FINALSTR
    end if
  else
    put it into FINALSTR
  end if
  put FINALSTR into line TARGETLINE of fld bg_fld_comments
end add_short_comment

on add_repair_job
  ask "Type in the job please...(60 char max)"
  put it into STRTOCNECK
  put empty into FINALSTR
  if length (STRTOCHECK) > 60 then
    answer "String too long...It will be trancated."
    with "OK" or "EDIT"
    if it is "OK" then
      put empty into NEWSTR
      repeat with i = 1 to 60
        put char i of STRTOCHECK after NEWSTR
      end repeat
      put NEWSTR into FINALSTR
    else
      put STRTOCHECK into it
      repeat until length(it) < 61
        ask"Edit it please." with it
        put it into FINALSTR
      end repeat
    end if
  else
    put STRTOCNECK into FINALSTR
  end if
  put "  " & FINALSTR & return after fld bg_fld_RER_maint
end add_repair_job

on remove_repair_job
  answer "Click on the job you want to remove !"
  wait until the mouseclick
  if line clickline() of fld bg_fld_REP_check is empty and
  line clickline() of fld bg_fld_RER_notes is empty and
  line clickline() of fld bg_fld_RER_to_log is empty and
  line clickline() of fld bg_fld_RER_days_fwd is empty then
    put quote & "Job #" & clickline() && "will be removed !" & quote
    into ASKSTRING
    answer ASKSTRING with "Cancel" or "Ok"
    if it is "Cancel" then
      exit remove_repair_job
    else
      delete line clickline() of fld bg_fld_rep_maint
      delete line clickline() of fld bg_fld_rep_check
      delete line clickline() of fld bg_fld_rep_notes
      delete line clickline() of fld bg_fld_rep_to_log
      delete line clickline() of fld bg_fld_rep_days_fwd
    end if
  else
    answer "Job may not be removed !"
  end if
end remove_repair_job
```

```
  if line TARGETLINE of fld bg_fld_rep_maint is not empty then
    put "\326" into line TARGETLINE of fld bg_fld_REP_check
  end if
end check_repair_job

on uncheck_repair_job
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_rep_maint is not empty then
    put empty into line TARGETLINE of fld bg_fld_REP_check
  end if
end uncheck_repair_job

on transfer_REP_to_log
  global TRANSFER_ME
  global WHEN_DONE
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_REP_check is "\326" then
    push card
    put line TARGETLINE of fld bg_fld_rep_maint into TRANSFER_ME
    delete char 1 of TRANSFER_ME
    put fld bg_fld_date into WHEN_DONE
    convert WHEN_DONE to short date
    go to stack log
    lock screen
    show fld click_to_set_rep_fld
    show bg button put_it_here
    show bg button exit_transfer_REP
    unlock screen
  else
    answer "Job must be completed to be transfered."
  end if
end transfer_REP_to_log

on go_REP_to_log
  push card
  go to stack log
end go_REP_to_log

on cancel_repair_job
  global TARGETLINE
  if line TARGETLINE of fld bg_fld_REP_maint is not empty then
    if line TARGETLINE of fld bg_fld_REP_check is not "\326" then
      put "-" into line TARGETLINE of fld bg_fld_REP_check
    else
      answer "You may not cancel a completed job."
    end if
  end if
end cancel_repair_job

on fill_note_repair_job
  global TARGETLINE
  global HOW_MANY_FIELDS
  lock screen
  if line TARGETLINE of fld bg_fld_note_id_repair is empty then
    --make new fld
    put the number of card fields into HOW_MANY_FIELDS --chk # of old flds
    put HOW_MANY_FIELDS + 1 into HOW_MANY_FIELDS --add one for new fld
    choose field tool
    drag from 242,184 to 506,314 with commandkey
    --set new field properties
    set the style of card field HOW_MANY_FIELDS to scrolling --set style
    set the textsize of card fld HOW_MANY_FIELDS to 9 --set font size
    set the textfont of card fld HOW_MANY_FIELDS to Geneva --set font
  else
    set the locktext of card field HOW_MANY_FIELDS to false
    show card field HOW_MANY_FIELDS
  end if
  --show new buttons
  show bg button bg_but_frame
  show bg button bg_but_print_note_repair
  show bg button bg_but_del_note_repair
  show bg button bg_but_close_note_repair
  --fill in info
  put "REP job #" & line TARGETLINE of fld bg_fld_REP_number
  && "                  " && the short date && "      -      "
  && the time & return after card fld HOW_MANY_FIELDS
  put "................................................................................"
  & return & return after card fld HOW_MANY_FIELDS
  --return to normal
  choose browse tool
  unlock screen
end fill_note_repair_job

on see_note_repair_job
  global TARGETLINE
  --connect to activator
  put line TARGETLINE of fld bg_fld_note_id_repair into it
  if it is not empty then
    show card fld id it
    --show buttons
    show bg button bg_but_frame
    show bg button bg_but_print_note_repair
    show bg button bg_but_del_note_repair
    show bg button bg_but_close_note_repair
  end if
end see_note_repair_job

 ***** BKGND #9, FIELD #1: bg_fld_job_compl_by *****
on mouseDown
  global TARGETLINE
  put CLICKLINE() into TARGETLINE
  activate_Top
end mouseDown

 ***** BKGND #9, FIELD #2: bg_fld_MRC_no *****
on mouseDown
  global TARGETLINE
  put empty into TARGETLINE
  put CLICKLINE() into TARGETLINE
  activate_Top
end mouseDown

 ***** BKGND #9, FIELD #3: bg_fld_comments *****
on mouseDown
  global TARGETLINE
  put empty into TARGETLINE
  put CLICKLINE() into TARGETLINE
  select line TARGETLINE of the target
  activate_Top
end mouseDown

 ***** BKGND #9, FIELD #11: bg_fld_rep_maint *****
on mouseDown
  global TARGETLINE
  put CLICKLINE() into TARGETLINE
  activate_Bottom
end mouseDown
```

```
 ***** BKGND #9, FIELD #20: bg_fld avail personel *****
on mousedown
  set locktext of me to false
  select line clickline() of me
  set locktext of me to true
  wait until the mouseclick
  if item 1 of the clickloc > 24 and item 1 of the clickloc < 222
  and item 2 of the clickloc > 91 and item 2 of the clickloc < 172 then
    put the selection into line NEWCLICKLINE() of fld bg_fld_job_compl_by
  else
    if item 1 of the clickloc > 226 and item 1 of the clickloc < 464
    and item 2 of the clickloc > 91 and item 2 of the clickloc < 281 then
      send mouseDown to me
    end if
  end if
end mousedown

 ***** BKGND #9, FIELD #21: bg_fld_tot_mh *****
on mouseDown
  global TARGETLINE
  global HOW_MANY_FIELDS
  global TEMPITEM
  global TEMPLINE
  global menu1
  global TARGETLINE
  put empty into HOW_MANY_FIELDS
  put empty into TARGETLINE
  put CLICKLINE() into TARGETLINE
  put "Check a Job Done" into menu1
  put return & "Delete a Job Checked"  after menu1
  put return & "Notes,Fill in a Note,See a Note,See all Notes" after menu1
  put return & "Add Short Comment,Job completed Succesfully.,Missing MRC card.,Tools not
 Available.,OTHER" after menu1
  put return & "Equipment Log" after menu1
  put return & "MRC Card" after menu1
  put return & "MIP Card" after menu1
  put return & "Personel,Assign jobs," after menu1
  get HPopupMenu(menu1,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theitem
    if theline = 3 and theitem = 2 then
      new_note
    end if
    if theline = 3 and theitem = 3 then
      see_note
    end if
    if theline = 3 and theitem = 4 then
      see_all_notes
    end if
    if theline = 4 then
      put theitem into TEMPITEM
      put theline into TEMPLINE
      add_short_comment
    end if
    if theline = 5 then
      lock screen
      push card
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to card DESTINATION of stack PMS
      put line 2 of fld data into it
      go to card it of stack log
      unlock screen
```

```
    if theline = 6 and theitem = 1 then
      lock screen
      push card
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to card DESTINATION of stack PMS
      unlock screen
    end if
    if theline = 7 and theitem = 1 then
      lock screen
      push card
      put line TARGETLINE of fld bg_fld_MRC_no into DESTINATION
      go to card DESTINATION of stack PMS
      put field "mip to go" into it
      go to card it
      unlock screen
    end if
    if theline = 8 and theitem = 2 then
      set cursor to 4
      lock screen
      show fld bg_fld_avail_personel
      show bg button "     Available Personel"
      show bg button bg_but_close_personel
      unlock screen
    end if
  end if --not zero
end mouseDown

on new_note
  global TARGETLINE
  global HOW_MANY_FIELDS
   lock screen
  --make new fld
  put the number of card fields into HOW_MANY_FIELDS --chk # of old flds
  put HOW_MANY_FIELDS + 1 into HOW_MANY_FIELDS --add one for new fld
  choose field tool
  drag from 242,184 to 506,314 with commandkey
  --set new field properties
  set the style of card field HOW_MANY_FIELDS to shadow --set style
  set the textsize of card fld HOW_MANY_FIELDS to 9 --set font size
  set the textfont of card fld HOW_MANY_FIELDS to Geneva --set font
  --show new buttons
  show bg button bg_but_frame
  show bg button bg_but_print_note
  show bg button bg_but_del_note
  show bg button bg_but_close_note
  --fill in info
  put line TARGETLINE of fld bg_fld_MRC_no
  && "                    " && the short date && "                    "
  && the time & return into line 1 of card fld HOW_MANY_FIELDS
  put "................................................................................
..."
  into line 2 of card fld HOW_MANY_FIELDS
  --return to normal
  choose browse tool
  unlock screen
  click at 248,213
end new_note
on see_note
  global HOW_MANY_FIELDS
  global TARGETLINE
  --connect to activator
  if it is not zero then
```

```
    -- put NEW_FLD_ID into line TARGETLINE of fld bg fld note_id
    --show buttons
    show bg button bg_but_frame
    show bg button bg_but_print_note
    show bg button bg_but_del_note
    show bg button bg_but_close_note
  end if
end see_note

on see_all_notes
  lock screen
  repeat with i=1 to the number of card fields
    if the visible of card fld i is true then
      answer "You mast close the note first..."
      exit see_all_notes
    end if
  end repeat
  hide bg button bg_but_frame
  hide bg button bg_but_print_note
  hide bg button bg_but_del_note
  hide bg button bg_but_close_note
   put empty into fld bg_fld_all_notes
  repeat with i=1 to the number of card fields
    put card fld i after fld bg_fld_all_notes
    repeat with ii=1 to 3
      put return after fld bg_fld_all_notes
    end repeat
  end repeat
  show fld bg_fld_all_notes
  --show buttons
  show bg button bg_but_frame_all_notes
  show bg button bg_but_print_note_all
  show bg button bg_but_close_note_all
  unlock screen
end see_all_notes

on add_short_comment
  global templtem
  global TEMPLINE
  global menul
  global TARGETLINE
  get item templtem of line TEMPLINE of menul
  if it is "OTHER" then
    set cursor to 4
    ask "Type in a SHORT comment..."
    put it into STRTOCHECK
    put empty into STRSTORAGE
    put empty into FINALSTR
    put 0 into CHARCOUNT
    put length(STRTOCHECK) into STRLENGTH
    put (STRLENGTH * 8) into STRLENGTH
    if STRLENGTH > 180 then
      answer "String too long...it will be trancated."
      with "CANCEL" or "OK"
      if it is "OK" then
        repeat until (CHARCOUNT > length(STRTOCHECK)) or
          (STRSTORAGE > 170)
          put CHARCOUNT + 1 into CHARCOUNT
          if charToNum of character CHARCOUNT of STRTOCHECK is 77¬
          or charToNum of character CHARCOUNT of STRTOCHECK is 87
          or charToNum of character CHARCOUNT of STRTOCHECK is 109
          or charToNum of character CHARCOUNT of STRTOCHECK is 119 then
            put (STRSTORAGE + 8) into STRSTORAGE
            put character CHARCOUNT of STRTOCHECK after FINALSTR
            next repeat
          end if
          if charToNum of character CHARCOUNT of STRTOCHECK is 73¬
          or charToNum of character CHARCOUNT of STRTOCHECK is 105¬
          or charToNum of character CHARCOUNT of STRTOCHECK is 106¬
          or charToNum of character CHARCOUNT of STRTOCHECK is 108 then
            put (STRSTORAGE + 3) into STRSTORAGE
            put character CHARCOUNT of STRTOCHECK after FINALSTR
            next repeat
          end if
          if (charToNum of character CHARCOUNT of STRTOCHECK >= 97¬
          and charToNum of character CHARCOUNT of STRTOCHECK <= 104) or¬
          (charToNum of character CHARCOUNT of STRTOCHECK >= 110¬
          and charToNum of character CHARCOUNT of STRTOCHECK <= 117) or¬
          charToNum of character CHARCOUNT of STRTOCHECK = 121
          or charToNum of character CHARCOUNT of STRTOCHECK = 122 then
            put (STRSTORAGE + 5) into STRSTORAGE
            put character CHARCOUNT of STRTOCHECK after FINALSTR
            next repeat
          end if
          put (STRSTORAGE + 6) into STRSTORAGE
          put character CHARCOUNT of STRTOCHECK after FINALSTR
        end repeat
        put FINALSTR & "..." into FINALSTR
      else
        exit add_short_comment
      end if
    else
      put STRTOCHECK into FINALSTR
    end if
  else
    put it into FINALSTR
    end if--STRTOCHECK is OTHER
    put FINALSTR into line TARGETLINE of fld bg_fld_comments
    end add_short_comment

 ***** BKGND #9, BUTTON #2: New Button *****
on mouseUp
  ask "Search for what ?"
  if it is empty or it is "CANCEL" then
    exit mouseUp
  else
    put it into FIND_WHAT
    put fld bg_fld_wc_da into TEMP_WC
    push card
    find FIND_WHAT
    if the result is not empty then
      beep
      answer "String not found"
      exit mouseUp
    else
      if fld bg_fld_wc_da is not TEMP_WC then
        beep
        answer "String not found"
        pop card
        exit mouseUp
      else
        exit mouseUp
      end if
    end if
  end if
end mouseUp

 ***** BKGND #9, BUTTON #4: Print_DA *****
```

```
on mouseUp
  set cursor to 4
  lock screen
  repeat with i = 1 to 5
    if the visible of button i is true then
      put i into STATUS
      exit repeat
    end if
  end repeat
  put fld bg_fld_job_compl_by into VAR1
  put fld bg_fld_MRC_no into VAR2
  put fld bg_fld_comments into VAR3
  put fld bg_fld_PMS_days_fwd into VAR4
  put fld bg_fld_to_log into VAR5
  put fld bg_fld_note_activator into VAR6
  put fld bg_fld_PMS_check into VAR7
  put fld bg_fld_rep_maint into VAR8
  put fld bg_fld_rep_days_fwd into VAR9
  put fld bg_fld_rep_to_log into VAR10
  put fld bg_fld_rep_notes into VAR11
  put fld bg_fld_rep_check into VAR12
  put fld bg_fld_date into VAR13
  put fld bg_fld_wc_DA into VAR14
  put fld calculated_mh into VAR15
  put fld completed into VAR16
  push card
  go to stack DA_report
  put VAR13 into fld date_report
  put VAR14 into fld WC_report
  put VAR1 into fld bg_fld_job_compl_by_report
  put VAR2 into fld bg_fld_MRC_no_report
  put VAR3 into fld bg_fld_comments_report
  put VAR4 into fld bg_fld_PMS_days_fwd_report
  put VAR5 into fld bg_fld_to_log_report
  put VAR6 into fld bg_fld_note_activator_report
  put VAR7 into fld bg_fld_PMS_check_report
  show bg button STATUS
  go next
  put VAR8 into fld bg_fld_rep_maint_report
  put VAR9 into fld bg_fld_rep_days_fwd report
  put VAR10 into fld bg_fld_log_rep_report
  put VAR11 into fld bg_fld_note_rep_report
  put VAR12 into fld bg_fld_rep_check_report
  put VAR15 into fld mh_report
  put VAR16 into fld completed_report
  put the date && "   " && the time into fld printed report
  open printing
  go first
  print all cards
  close printing
  hide bg button STATUS
  pop card
  set cursor to 1
  unlock screen
end mouseUp

 ***** BKGND #9, BUTTON #5: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #9, BUTTON #6: down *****
```

```
  add 72 to it
  set the lockscreen to true
  set the scroll of fld bg_fld_job_compl_by to it
  set the scroll of fld bg_fld_MRC_no to it
  set the scroll of fld bg_fld_comments to it
  set the scroll of fld bg_fld_PMS_days_fwd to it
  set the scroll of fld bg_fld_to_log to it
  set the scroll of fld bg_fld_note_activator to it
  set the scroll of fld bg_fld_PMS_check to it
  set the lockscreen to false
end mouseUp

 ***** BKGND #9, BUTTON #7: up *****
on mouseUp
  put the scroll of field bg_fld_PMS_check into it
  subtract 72 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of fld bg_fld_job_compl_by to it
  set the scroll of fld bg_fld_MRC_no to it
  set the scroll of fld bg_fld_comments to it
  set the scroll of fld bg_fld_PMS_days_fwd to it
  set the scroll of fld bg_fld_to_log to it
  set the scroll of fld bg_fld_note_activator to it
  set the scroll of fld bg_fld_PMS_check to it
  set the lockscreen to false
end mouseUp

 ***** BKGND #9, BUTTON #8: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #9, BUTTON #13: down *****
on mouseUp
  put the scroll of field bg_fld_REP_check into it
  add 72 to it
  set the lockscreen to true
  set the scroll of fld bg_fld_rep_maint to it
  set the scroll of fld bg_fld_REP_days_fwd to it
  set the scroll of fld bg_fld_REP_to_log to it
  set the scroll of fld bg_fld_REP_notes to it
  set the scroll of fld bg_fld_REP_check to it
  set the scroll of fld bg_fld_REP_number to it
  set the lockscreen to false
end mouseUp

 ***** BKGND #9, BUTTON #14: up *****
on mouseUp
  put the scroll of field bg_fld_REP_check into it
  subtract 72 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of fld bg_fld_rep_maint to it
  set the scroll of fld bg_fld_REP_days_fwd to it
  set the scroll of fld bg_fld_REP_to_log to it
  set the scroll of fld bg_fld_REP_notes to it
  set the scroll of fld bg_fld_REP_check to it
  set the scroll of fld bg_fld_REP_number to it
  set the lockscreen to false
end mouseUp
```

```
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #9, BUTTON #24: New Button *****
on mouseDown
  put "Click on the line you are interested to act on" into menu1
  get NPopupMenu(menu1,0,the mouseV,the mouseH)
end mouseDown

 ***** BKGND #9, BUTTON #25: New Button *****
on mouseDown
  put "Click on the line you are interested on" into menu1
  get NPopupMenu(menu1,0,the mouseV,the mouseH)
end mouseDown

 ***** BKGND #9, BUTTON #26: New Button *****
on mouseDown
  put "Click on the line you are interested on" into menu1
  get NPopupMenu(menu1,0,the mouseV,the mouseH)
end mouseDown

 ***** BKGND #9, BUTTON #27: New Button *****
on mouseDown
  put "Click on the line you are interested on" into menu1
  get NPopupMenu(menu1,0,the mouseV,the mouseH)
end mouseDown

 ***** BKGND #9, BUTTON #29: bg_but_close_note *****
on mousedown
  put "Close Note" into menu1
  get NPopupMenu(menu1,0,146,241)
end mousedown

on mouseUp
  global NOW_MANY_FIELDS
  global TARGETLINE
  if the locktext of card fld HOW_MANY_FIELDS is false then
    set the locktext of card fld NOW_MANY_FIELDS to true
    put line TARGETLINE of fld bg_fld_job_compl_by into SIGN_IT
    ask "Is this the name you want to sign with ?" with SIGN_IT
    put it into SIGN_IT
    --connect to activator
    put the id of card fld NOW_MANY_FIELDS into NEW_FLD_ID --get id
    put NEW_FLD_ID into line TARGETLINE of fld bg_fld_note_id
    put "\326" into line TARGETLINE of fld bg_fld_note_activator
    put "See Attached Notes." into line TARGETLINE of fld bg_fld_comments
    put SIGN_IT after card fld NOW_MANY_FIELDS --sign it
    put return & "_______________________________________"
    after card fld NOW_MANY_FIELDS
  end if
  --shide new fld
  hide card fld HOW_MANY_FIELDS
  --show new buttons
  hide bg button bg_but_frame
  hide bg button bg_but_print_note
  hide bg button bg_but_del_note
  hide bg button bg_but_close_note
end mouseUp

 ***** BKGND #9, BUTTON #30: bg_but_print_note *****
on mouseDown
  put "Print Note" into menu1
  get NPopupMenu(menu1,0,146,242)
end mouseDown

on mouseUp
  repeat with i = 1 to the number of card fields
    if the visible of card field i is true then
      put i into THIS_FIELD
      put quote & "card field" && THIS_FIELD & quote into it
      PrintField it
      exit repeat
    end if
  end repeat
end mouseUp

 ***** BKGND #9, BUTTON #31: bg_but_del_note *****
on mouseDown
  put "Discard Note" into menu1
  get NPopupMenu(menu1,0,146,234)
end mouseDown

on mouseUp
  global TARGETLINE
  answer "This opperation will delete the entire note" with "Discard"
  or "Cancel"
  if it is "Discard" then
    lock screen
    choose field tool
    click at 336,236
    type "x" with commandkey
    hide bg button bg_but_frame
    hide bg button bg_but_print_note
    hide bg button bg_but_del_note
    hide bg button bg_but_close_note
    put empty into line TARGETLINE of fld bg_fld_note_id
    put empty into line TARGETLINE of fld bg_fld_note_activator
    put empty into line TARGETLINE of fld bg_fld_comments
    choose browse tool
    unlock screen
  else
    exit mouseUp
  end if
end mouseUp

 ***** BKGND #9, BUTTON #32: bg_but_print_note_all *****
on mouseDown
  put "Print Note" into menu1
  get NPopupMenu(menu1,0,17,111)
end mouseDown

on mouseUp
  PrintField ("field bg_fld_all_notes")
end mouseUp

 ***** BKGND #9, BUTTON #33: bg_but_close_note_all *****
on mouseDown
  put "Close Note" into menu1
  get NPopupMenu(menu1,0,17,111)
end mouseDown

on mouseUp
  hide fld bg_fld_all_notes
  --hide buttons
  hide bg button bg_but_frame_all_notes
  hide bg button bg_but_print_note_all
  hide bg button bg_but_close_note_all
```

```
end mouseUp

 ***** BKGND #9, BUTTON #35: bg_but_close_personel *****
on mouseDown
  put "Hide Personel" into menu1
  get HPopupMenu(menu1,0,55,225)
end mouseDown

on mouseUp
  lock screen
  hide fld bg_fld_avail_personel
  hide bg button "     Available Personel"
  hide bg button bg_but_close_personel
  unlock screen
end mouseUp

 ***** BKGND #9, BUTTON #38: bg_but_del_note_repair *****
on mouseDown
  put "Discard Note" into menu1
  get HPopupMenu(menu1,0,146,234)
end mouseDown

on mouseUp
  global TARGETLINE
    answer "This opperation will delete the entire note" with "Discard"
  or "Cancel"
  if it is "Discard" then
    lock screen
    choose field tool
    click at 336,236
    type "x" with commandkey
    hide bg button bg_but_frame
    hide bg button bg_but_print_note_repair
    hide bg button bg_but_del_note_repair
    hide bg button bg_but_close_note_repair
    put empty into line TARGETLINE of fld bg_fld_note_id_repair
    put empty into line TARGETLINE of fld bg_fld_REP_notes
    choose browse tool
    unlock screen
  else
    exit mouseUp
  end if
end mouseUp

 ***** BKGND #9, BUTTON #39: bg_but_print_note_repair *****
on mouseDown
  put "Print Note" into menu1
  get HPopupMenu(menu1,0,146,242
end mouseDown

on mouseUp
  repeat with i = 1 to the number of card fields
    if the visible of card field i is true then
      put i into THIS_FIELD
      put quote & "card field" && THIS_FIELD & quote into it
      PrintField it
      exit repeat
    end if
  end repeat
end mouseUp

 ***** BKGND #9, BUTTON #40: bg_but_close_note_repair *****
on mousedown
  get HPopupMenu(menu1,0,146,241)
end mousedown

on mouseUp
  global HOW_MANY_FIELDS
  global TARGETLINE
  if the locktext of card fld HOW_MANY_FIELDS is false then
    set the locktext of card fld HOW_MANY_FIELDS to true
    put line TARGETLINE of fld bg_fld_job_compl_by into SIGN_IT
    ask "Is this the name you want to sign with ?" with SIGN_IT
    put it into SIGN_IT
    --connect to activator
    put the id of card fld HOW_MANY_FIELDS into NEW_FLD_ID --get id
    put NEW_FLD_ID into line TARGETLINE of fld bg_fld_note_id_repair
    put "\326" into line TARGETLINE of fld bg_fld_REP_notes
    put SIGN_IT after card fld HOW_MANY_FIELDS --sign it
    put return & "_______________________________________"
    after card fld HOW_MANY_FIELDS
  end if
  --shide new fld
  hide card fld HOW_MANY_FIELDS
  --show new buttons
  hide bg button bg_but_frame
  hide bg button bg_but_print_note_repair
  hide bg button bg_but_del_note_repair
  hide bg button bg_but_close_note_repair
end mouseUp

 ***** BKGND #10, BUTTON #1: WC Access *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- hide all check boxes to prepare screen to show the correct ones
  repeat with i = 1 to the number of buttons
    hide button i
  end repeat
  -- inform the user what mode he is on
  Put the short name of me into fld access_to_what
  -- show appropriate check boxes for the requested mode
  show card button "Engineering"
  show card button "E-1"
  show card button "EM01"
  show card button "EM06"
  show card button "E-2"
  show card button "EM02"
  show card button "EM03"
  show card button "EP09"
  show card button "E-3"
  show card button "EM04"
  show card button "EM05"
  show card button "S-1"
  show card button "SS02"
  show card button "Support"
  show card button "S-2"
  show card button "SS06"
  show card button "S-3"
  show card button "SS03"
  show card button "SS05"
  show card button "SS07"
  show card button "S-4"
  show card button "SS01"
  show card button "SS04"
  show card button "SC-1"
  show card button "OC01"
  show card button "SC-2"
  show card button "OC02"
  show card button "SC-3"
  show card button "OC03"
  show card button "CS-1"
  show card button "CS-2"
  show card button "CS-3"
  show card button "CS-4"
  show card button "CS01"
  show card button "CS02"
  show card button "CS03"
  show card button "CS04"
  show card button "CS05"
  show card button "CS06"
  show card button "CS07"
  show card button "CS08"
  show card button "CS09"
  -- restore screen drawing to the new situation
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #10, BUTTON #2: Equip. Log *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- hide all check boxes to prepare screen to show the correct ones
  repeat with i = 1 to the number of buttons
    hide button i
  end repeat
  -- inform the user what mode he is on
  Put the short name of me into fld access_to_what
  -- show appropriate check boxes for the requested mode
  show card button "Enter Today's Date"
  show card button "Mfctr Data"
  show card button "Modify mfctr Data"
  show card button "Publications"
  show card button "Go To Pub"
  show card button "Modify Pubs"
  show card button "Drawing numbers"
  show card button "Go To Drawing"
  show card button "Modify Drawings"
  show card button "Date"
  show card button "Dial Mfctr"
  show card button "Remove Date"
  show card button "Enter Other Date"
  show card button "Remarks"
  show card button "Add Data"
  show card button "Modify remarks"
  -- restore screen to new situation
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #10, BUTTON #3: DA Cards *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- hide all check boxes to prepare screen to show the correct ones
  repeat with i = 1 to the number of buttons
    hide button i
  end repeat
  -- inform the user what mode he is on
  Put the short name of me into fld access_to_what
  -- show appropriate check boxes for the requested mode
  show card button "Done"
  show card button "Notes"
  show card button "Fill"
  show card button "Repair Maintenance"
  show card button "Check job"
  show card button "PMS Menus"
  show card button "Cancel check"
  show card button "Cancel PMS job"
  show card button "See"
  show card button "See all"
  show card button "go DA"
  show card button "Equipment log DA"
  show card button "Go to log"
  show card button "Add job to log"
  show card button "Comment"
  show card button "MRC card"
  show card button "MIP card"
  show card button "Personel"
  show card button "Assign job"
  show card button "Rmv person"
  show card button "Arrange repair job"
  show card button "Add a job"
  show card button "Delete a job"
  show card button "Transfer a job"
  show card button "Check repair job"
  show card button "Done repair"
  show card button "Cancel check repair"
  show card button "Cancel repair job"
  -- restore screen to new situation
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #10, BUTTON #4: Toolbox *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- hide all check boxes to prepare screen to show the correct ones
  repeat with i = 1 to the number of buttons
    hide button i
  end repeat
  -- inform the user what mode he is on
  Put the short name of me into fld access_to_what
  -- show appropriate check boxes for the requested mode
  show card button "Toolbox Access"
  show card button "Toolbox"
  show card button "Add more MIP cards"
  show card button "Convert MIP-MRC file to stack"
  show card button "Empty PMS database"
  show card button "MIP-MRC cards in PMS database"
  show card button "Delete a MIP set"
  show card button "WC responsibility"
  -- restore screen to new situation
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #10, BUTTON #5: General *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- hide all check boxes to prepare screen to show the correct ones
  repeat with i = 1 to the number of buttons
    hide button i
  end repeat
  -- Inform the user what mode he is on
  Put the short name of me into fld access_to_what
  -- show appropriate check boxes for the requested mode
  show card button "General"
  show card button "Password"
  show card button "Add new member"
  show card button "Remove a member"
  show card button "Change member's access"
  show card button "Change my password"
  show card button "Spot checks"
  show card button "For entire ship"
  show card button "For one WC"
  show card button "Print"
  show card button "Today's DA cards - all"
  show card button "Today's DA card - 1 WC"
  show card button "MIP card print"
  show card button "MRC card print"
  show card button "Log card print"
  show card button "Reports & messages"
  show card button "% done diagram - 1 WC"
  show card button "% done diagram - all WC"
  show card button "Navy message"
  show card button "Ext. rep req"
  show card button "Go"
  show card button "MIP card go"
  show card button "MRC card go"
  show card button "Log card go"
  show card button "Bar coded label"
  show card button "Schedule"
  show card button "Next quarter"
  show card button "Equipment log"
  show card button "Add entry"
  show card button "Del entry"
  -- restore screen to new situation
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #10, BUTTON #6: Set_me *****
on mouseUp
  -- check whether it is a new card or a revised one
  If the short name of this stack is "Password stack" then
    -- if it is a revised one...
    go to card 2 of atack "PMS_driver"
  else
    -- if it is a new member...
    -- lock the acreen to make things faster
    lock screen
    -- remove it from the PMS_driver
    doMenu "Cut Card"
    -- go to the password stack
    go to last card of stack "Password stack"
    -- place it there
    doMenu "Paste Card"
    -- compact the stack to reduce its size
    doMenu "Compact Stack"
    --return to the PMS_driver
    go to card 2 of stack "PMS_driver"
    -- reset the screen
```

```
  end if
end mouseUp

 ***** BKGND #10, BUTTON #7: delete_me *****
on mouseUp
  -- ask the user to verify that he really wants to remove this card
  answer "Are you sure ?" with "Ok" or "Cancel"
  -- if he does...
  If it is "Ok" then
    -- lock the screen to make things ran faster
    lock screen
    -- throw the card away
    doMenu "Delete Card"
    -- compact the stack to give back to the system the space taken from this card
    doMenu "Compact Stack"
    --return to the driver
    go to card 2 of stack "PMS_driver"
    -- reset the screen
    unlock screen
  else
    -- forget it...
    exit mouseUp
  end if
end mouseUp

 ***** BKGND #10, BUTTON #8: print_me *****
on mouseUp
  -- suspend screen drawing to speed up execution
  lock screen
  -- show General menus
  send mouseup to bg button "General"
  -- ...and print them
  print this card
  -- show WC Access menus
  send mouseup to bg button "WC Access"
  -- ...and print them
  print this card
  -- show "Equip. Log" menus
  send mouseup to bg button "Equip. Log"
  -- ...and print them
  print this card
  -- show "DA Cards" menus
  send mouseup to bg button "DA Cards"
  -- ...and print them
  print this card
  -- show Toolbox menus
  send mouseup to bg button "Toolbox"
  -- ...and print them
  print this card
  -- show General menus to restore original condition
  send mouseup to bg button "General"
  -- restore screen
  unlock screen
  -- exit event
end mouseUp


 ***** BKGND #10, BUTTON #9: do_it *****
on mouseUp
  global DRIV_GENERAL_MENU
  global DRIV_ENGINEERING_MENU
  global DRIV_SUPPORT_MENU
  global DRIV_SHIP_CONTROL_MENU
global LOG_MFCTR_DATA_MENU
global LOG_PUBS_MENU
global LOG_DRWG_NUM_MENU
global LOG_DATE_MENU
global LOG_REMARK_MENU
global DA_PMS_MENU
global DA_REP_MAINT_MENU
-- check if the hilite of card button "MRC card print" is true to
-- make this menu option available
if the hilite of card button "MRC card print" then
  delete char 1 of item 5 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Password" is true to
-- make this menu option available
if the hilite of card button "Password" then
  delete char 1 of item 1 of line 1 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Add new member" is true to
-- make this menu option available
if the hilite of card button "Add new member" then
  delete char 1 of item 2 of line 1 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Remove a member" is true to
-- make this menu option available
if the hilite of card button "Remove a member" then
  delete char 1 of item 3 of line 1 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Change member's access" is true to
-- make this menu option available
if the hilite of card button "Change member's access" then
  delete char 1 of item 4 of line 1 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Change my password" is true to
-- make this menu option available
if the hilite of card button "Change my password" then
  delete char 1 of item 5 of line 1 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Spot checks" is true to
-- make this menu option available
if the hilite of card button "Spot checks" then
  delete char 1 of item 1 of line 4 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "For entire ship" is true to
-- make this menu option available
if the hilite of card button "For entire ship" then
  delete char 1 of item 2 of line 4 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "For one WC" is true to
-- make this menu option available
if the hilite of card button "For one WC" then
  delete char 1 of item 3 of line 4 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Print" is true to
-- make this menu option available
if the hilite of card button "Print" then
  delete char 1 of item 1 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Today's DA cards - all" is true to
-- make this menu option available
if the hilite of card button "Today's DA cards - all" then
  delete char 1 of item 2 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Today's DA card - 1 WC" is true to
-- make this menu option available
if the hilite of card button "Today's DA card - 1 WC" then
  delete char 1 of item 3 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "MIP card print" is true to
-- make this menu option available
if the hilite of card button "MIP card print" then
  delete char 1 of item 4 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Log card print" is true to
-- make this menu option available
if the hilite of card button "Log card print" then
  delete char 1 of item 6 of line 6 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Reports & messages" is true to
-- make this menu option available
if the hilite of card button "Reports & messages" then
  delete char 1 of item 1 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "% done diagram - 1 WC" is true to
-- make this menu option available
if the hilite of card button "% done diagram - 1 WC" then
  delete char 1 of item 2 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "% done diagram - all WC" is true to
-- make this menu option available
if the hilite of card button "% done diagram - all WC" then
  delete char 1 of item 3 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Navy message" is true to
-- make this menu option available
if the hilite of card button "Navy message" then
  delete char 1 of item 5 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Ext. rep req" is true to
-- make this menu option available
if the hilite of card button "Ext. rep req" then
  delete char 1 of item 6 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Schedule" is true to
-- make this menu option available
if the hilite of card button "Schedule" then
  delete char 1 of item 1 of line 2 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Next quarter" is true to
-- make this menu option available
if the hilite of card button "Next quarter" then
  delete char 1 of item 2 of line 2 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Go" is true to
-- make this menu option available
if the hilite of card button "Go" then
  delete char 1 of item 1 of line 5 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "MRC card go" is true to
-- make this menu option available
if the hilite of card button "MRC card go" then
  delete char 1 of item 2 of line 5 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "MIP card go" is true to
-- make this menu option available
if the hilite of card button "MIP card go" then
  delete char 1 of item 3 of line 5 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "navy message" is true to
-- make this menu option available
if the hilite of card button "navy message" then
  delete char 1 of item 4 of line 3 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Log card go" is true to
-- make this menu option available
if the hilite of card button "Log card go" then
  delete char 1 of item 4 of line 5 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Equipment log" is true to
-- make this menu option available
if the hilite of card button "Equipment log" then
  delete char 1 of item 1 of line 8 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Add entry" is true to
-- make this menu option available
if the hilite of card button "Add entry" then
  delete char 1 of item 2 of line 8 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "Del entry" is true to
-- make this menu option available
if the hilite of card button "Del entry" then
  delete char 1 of item 3 of line 8 of DRIV_GENERAL_MENU
end if
-- check if the hilite of card button "E-1" is true to
-- make this menu option available
if the hilite of card button "E-1" then
  delete char 1 of item 1 of line 1 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "EM01" is true to
-- make this menu option available
if the hilite of card button "EM01" then
  delete char 1 of item 2 of line 1 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "EM06" is true to
-- make this menu option available
if the hilite of card button "EM06" then
  delete char 1 of item 3 of line 1 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "E-2" is true to
-- make this menu option available
if the hilite of card button "E-2" then
  delete char 1 of item 1 of line 2 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "EM02" is true to
-- make this menu option available
if the hilite of card button "EM02" then
  delete char 1 of item 2 of line 2 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "EM03" is true to
-- make this menu option available
if the hilite of card button "EM03" then
  delete char 1 of item 3 of line 2 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "E-3" is true to
-- make this menu option available
if the hilite of card button "E-3" then
  delete char 1 of item 1 of line 3 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "EM04" is true to
-- make this menu option available
if the hilite of card button "EM04" then
  delete char 1 of item 2 of line 3 of DRIV_ENGINEERING_MENU
```

```
-- check if the hilite of card button "EM05" is true to
-- make this menu option available
if the hilite of card button "EM05" then
  delete char 1 of item 3 of line 3 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "ER09" is true to
-- make this menu option available
if the hilite of card button "ERD9" then
  delete char 1 of item 4 of line 3 of DRIV_ENGINEERING_MENU
end if
-- check if the hilite of card button "S-1" is true to
-- make this menu option available
if the hilite of card button "S-1" then
  delete char 1 of item 1 of line 1 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS02" is true to
-- make this menu option available
if the hilite of card button "SS02" then
  delete char 1 of item 2 of line 1 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "S-2" is true to
-- make this menu option available
if the hilite of card button "S-2" then
  delete char 1 of item 1 of line 2 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS06" is true to
-- make this menu option available
if the hilite of card button "SS06" then
  delete char 1 of item 2 of line 2 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "S-3" is true to
-- make this menu option available
if the hilite of card button "S-3" then
  delete char 1 of item 1 of line 3 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS03" is true to
-- make this menu option available
if the hilite of card button "SS03" then
  delete char 1 of item 2 of line 3 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS05" is true to
-- make this menu option available
if the hilite of card button "SSD5" then
  delete char 1 of item 3 of line 3 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS07" is true to
-- make this menu option available
if the hilite of card button "SSD7" then
  delete char 1 of item 4 of line 3 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "S-4" is true to
-- make this menu option available
if the hilite of card button "S-4" then
  delete char 1 of item 1 of line 4 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS01" is true to
-- make this menu option available
if the hilite of card button "SS01" then
  delete char 1 of item 2 of line 4 of DRIV_SUPPORT_MENU
end if
-- check if the hilite of card button "SS04" is true to
-- make this menu option available
```

```
end if
-- check if the hilite of card button "SC-1" is true to
-- make this menu option available
if the hilite of card button "SC-1" then
  delete char 1 of item 1 of line 1 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "OC01" is true to
-- make this menu option available
if the hilite of card button "OC01" then
  delete char 1 of item 2 of line 1 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "SC-2" is true to
-- make this menu option available
if the hilite of card button "SC-2" then
  delete char 1 of item 1 of line 2 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "OC02" is true to
-- make this menu option available
if the hilite of card button "OC02" then
  delete char 1 of item 2 of line 2 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "SC-3" is true to
-- make thia menu option available
if the hilite of card button "SC-3" then
  delete char 1 of item 1 of line 3 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "OC03" is true to
-- make thia menu option available
if the hilite of card button "OC03" then
  delete char 1 of item 2 of line 3 of DRIV_SHIP_CONTROL_MENU
end if
-- check if the hilite of card button "CS-1" is true to
-- make this menu option available
if the hilite of card button "CS-1" then
  delete char 1 of item 1 of line 1 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS01" is true to
-- make thia menu option available
if the hilite of card button "CS01" then
  delete char 1 of item 2 of line 1 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS02" is true to
-- make this menu option available
if the hilite of card button "CS02" then
  delate char 1 of item 3 of line 1 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS-2" is true to
-- make this menu option available
if the hilite of card button "CS-2" then
  delete char 1 of item 1 of line 2 of DRIV_COMBAT_SYS_MENU
end if
-- chack if the hilite of card button "CS03" is true to
-- make this manu option available
if tha hilite of card button "CS03" then
  delete char 1 of item 2 of line 2 of DRIV_COMBAT_SYS_MENU
end if
-- chack if the hilite of card button "CS-3" is true to
-- make this menu option available
if the hilite of card button "CS-3" then
  delete char 1 of item 1 of line 3 of DRIV_COMBAT_SYS_MENU
end if
-- check if tha hilite of card button "CS04" ia true to
-- make this menu option available
if the hilita of card button "CS04" then
  delete char 1 of item 2 of line 3 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS05" ia true to
-- make this menu option available
if the hilite of card button "CS05" then
  delete char 1 of item 3 of line 3 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS06" is true to
-- make this menu option available
if the hilite of card button "CS06" then
  delete char 1 of item 4 of line 3 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS-4" ia true to
-- make this menu option available
if the hilite of card button "CS-4" then
  delete char 1 of item 1 of line 4 of DRIV_COMBAT_SYS_MENU
end if
-- chack if the hilite of card button "CS07" is true to
-- make this menu option available
if the hilite of card button "CS07" then
  delete char 1 of item 2 of line 4 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS08" ia true to
-- make this menu option available
if the hilite of card button "CS08" then
  delete char 1 of item 3 of line 4 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "CS09" is true to
-- make this menu option available
if the hilite of card button "CS09" then
  delete char 1 of item 4 of line 4 of DRIV_COMBAT_SYS_MENU
end if
-- check if the hilite of card button "Enter today's date" is true to
-- make this menu option available
if the hilite of card button "Enter today'a date" then
  delete char 1 of item 1 of line 1 of LOG_DATE_MENU
end if
-- check if the hilite of card button "Go to pub" is true to
-- make this menu option available
if the hilite of card button "Go to pub" then
  delete char 1 of item 1 of line 1 of LOG_PUBS_MENU
end if
-- check if the hilite of card button "Modify puba" is true to
-- make this menu option available
if the hilite of card button "Modify pubs" then
  delete char 1 of item 1 of line 2 of LOG_PUBS_MENU
end if
-- check if the hilite of card button "Go to drawing" ia true to
-- make this menu option available
if the hilite of card button "Go to drawing" then
  delete char 1 of item 1 of line 1 of LOG_DRWG_NUM_MENU
end if
-- check if the hilite of card button "Modify drawinga" is true to
-- make this menu option available
if the hilite of card button "Modify drawings" then
  delete char 1 of item 1 of line 2 of LOG_DRWG_NUM_MENU
end if
-- check if the hilite of card button "Dial mfctr" ia true to
-- make this menu option available
if the hilite of card button "Dial mfctr" then
  delete char 1 of item 1 of line 1 of LOG_MFCTR_DATA_MENU
end if
-- check if the hilite of card button "Remove date" ia true to
-- make this menu option available
```

```
if the hilite of card button "Remove date" then
  delete char 1 of item 1 of line 2 of LOG_DATE_MENU
end if
-- check if the hilite of card button "Enter other date" is true to
-- make this menu option available
if the hilite of card button "Enter other date" then
  delete char 1 of item 1 of line 3 of LOG_DATE_MENU
end if
-- check if the hilite of card button "Add data" is true to
-- make this menu option available
if the hilite of card button "Add data" then
  delete char 1 of item 1 of line 1 of LOG_REMARK_MENU
end if
-- check if the hilite of card button "Modify remarks" is true to
-- make this menu option available
if the hilite of card button "Modify remarks" then
  delete char 1 of item 1 of line 2 of LOG_REMARK_MENU
end if
-- check if the hilite of card button "Modify mfctr data" is true to
-- make this menu option available
if the hilite of card button "Modify mfctr data" then
  delete char 1 of item 1 of line 2 of LOG_MFCTR_DATA_MENU
end if
-- check if the hilite of card button "Check job" is true to
-- make this menu option available
if the hilite of card button "Check job" then
  delete char 1 of item 1 of line 1 of DA_PMS_MENU
end if
-- check if the hilite of card button "Done" is true to
-- make this menu option available
if the hilite of card button "Done" then
  delete char 1 of item 2 of line 1 of DA_PMS_MENU
end if
-- check if the hilite of card button "Cancel check" is true to
-- make this menu option available
if the hilite of card button "Cancel check" then
  delete char 1 of item 3 of line 1 of DA_PMS_MENU
end if
-- check if the hilite of card button "Cancel PMS job" is true to
-- make this menu option available
if the hilite of card button "Cancel PMS job" then
  delete char 1 of item 4 of line 1 of DA_PMS_MENU
end if
-- check if the hilite of card button "Arrange repair job" is true to
-- make this menu option available
if the hilite of card button "Arrange repair job" then
  delete char 1 of item 1 of line 1 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Add a job" is true to
-- make this menu option available
if the hilite of card button "Add a job" then
  delete char 1 of item 2 of line 1 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Delete a job" is true to
-- make this menu option available
if the hilite of card button "Delete a job" then
  delete char 1 of item 3 of line 1 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Transfer a job" la true to
-- make this menu option available
if the hilite of card button "Transfer a job" then
  delete char 1 of item 4 of line 1 of DA_REP_MAINT_MENU
end if
```

```
-- make this menu option available
if the hilite of card button "Notes" then
  delete char 1 of item 1 of line 2 of DA_PMS_MENU
end if
-- check if the hilite of card button "Fill" is true to
-- make this menu option available
if the hilite of card button "Fill" then
  delete char 1 of item 2 of line 2 of DA_PMS_MENU
end if
-- check if the hilite of card button "See" is true to
-- make this menu option available
if the hilite of card button "See" then
  delete char 1 of item 3 of line 2 of DA_PMS_MENU
end if
-- check if the hilite of card button "See all" is true to
-- make this menu option available
if the hilite of card button "See all" then
  delete char 1 of item 4 of line 2 of DA_PMS_MENU
end if
-- check if the hilite of card button "Equipment log DA" is true to
-- make this menu option available
if the hilite of card button "Equipment log" then
  delete char 1 of item 1 of line 3 of DA_PMS_MENU
end if
-- check if the hilite of card button "Go to log" is true to
-- make this menu option available
if the hilite of card button "Go to log" then
  delete char 1 of item 2 of line 3 of DA_PMS_MENU
end if
-- check if the hilite of card button "Add job to log" is true to
-- make this menu option available
if the hilite of card button "Add job to log" then
  delete char 1 of item 3 of line 3 of DA_PMS_MENU
end if
-- check if the hilite of card button "Check repair job" is true to
-- make this menu option available
if the hilite of card button "Check repair job" then
  delete char 1 of item 1 of line 2 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Done repair" is true to
-- make thla menu option available
if the hilite of card button "Done repair" then
  delete char 1 of item 2 of line 2 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Cancel check repair" is true to
-- make thla menu option available
if the hilite of card button "Cancel check repair" then
  delete char 1 of item 3 of line 2 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Cancel repair job" la true to
-- make this menu option available
if the hilite of card button "Cancel repair job" then
  delete char 1 of item 4 of line 2 of DA_REP_MAINT_MENU
end if
-- check if the hilite of card button "Peraonel" is true to
-- make thla menu option available
if the hilite of card button "Personel" then
  delete char 1 of item 1 of line 6 of DA_PMS_MENU
end if
-- check if the hilite of card button "Assign job" is true to
-- make this menu option available
if the hilite of card button "Asaign job" then
  delete char 1 of item 2 of line 6 of DA_PMS_MENU
  -- check if the hilite of card button "Rmv person" is true to
  -- make this menu option available
  if the hilite of card button "Rmv person" then
    delete char 1 of item 3 of line 6 of DA_PMS_MENU
  end if
  -- check if the hilite of card button "Comment" is true to
  -- make this menu option available
  if the hilite of card button "Comment" then
    delete char 1 of item 1 of line 5 of DA_PMS_MENU
  end if
  -- check if the hilite of card button "MRC card" is true to
  -- make this menu option available
  if the hilite of card button "MRC card" then
    delete char 1 of item 2 of line 4 of DA_PMS_MENU
  end if
  -- check if the hilite of card button "MIP card" is true to
  -- make this menu option available
  if the hilite of card button "MIP card" then
    delete char 1 of item 3 of line 4 of DA_PMS_MENU
  end if
  -- check if the hilite of card button "Go DA" is true to
  -- make this menu option available
  if the hilite of card button "Go DA" then
    delete char 1 of item 1 of line 4 of DA_PMS_MENU
  end if
  -- check if the hilite of card button "Toolbox" is true to
  -- make this menu option available
  if the hilite of card button "Toolbox" then
    delete char 1 of item 1 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "Add more MIP cards" is true to
  -- make this menu option available
  if the hilite of card button "Add more MIP cards" then
    delete char 1 of item 2 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "Convert MIP-MRC file to stack" is true to
  -- make this menu option available
  if the hilite of card button "Convert MIP-MRC file to stack" then
    delete char 1 of item 3 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "Empty PMS database" is true to
  -- make this menu option available
  if the hilite of card button "Empty PMS database" then
    delete char 1 of item 4 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "MIP-MRC cards in PMS database" is true to
  -- make this menu option available
  if the hilite of card button "MIP-MRC cards in PMS database" then
    delete char 1 of item 5 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "Delete a MIP set" is true to
  -- make this menu option available
  if the hilite of card button "Delete a MIP set" then
    delete char 1 of item 6 of line 7 of DRIV_GENERAL_MENU
  end if
  -- check if the hilite of card button "WC responsibility" is true to
  -- make this menu option available
  if the hilite of card button "WC responsibility" then
    delete char 1 of item 7 of line 7 of DRIV_GENERAL_MENU
  end if
  set the lockMessages to true
  go to card PMS_driver_crd1 of stack "PMS_driver"
  unlock screen
end mouseUp
```

```
 ***** BKGND #12, BUTTON #3: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #12, BUTTON #4: New Button *****
on mouseUp
  if the id of this card = the id of last card  then
    answer "This is the last card of this Quarter."
  else
    go next
  end if
end mouseUp

 ***** BKGND #12, BUTTON #5: New Button *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #12, BUTTON #6: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #12, BUTTON #7: bg_but_create_DA_cards *****
on mouseUp

  answer "Schedule PMS jobs for this quarter ?" with
  "Cancel" or "OK"
  if it is "OK" then
    answer "This operation takes a couple of hours..." with "Cancel" or "Continue"
    if it is "Cancel" then
      exit mouseUp
    else
      answer "Automatic rescheduling active ?" with "Active" or "Inactive"
      if it is "Active" then
        put true into RESCH
      else
        put false into RESCN
      end if
      show card fld show_shut_down
      wait until the mouseClick
      hide card fld show_shut_down
    end if
    push card
    -- Create new stack with automatic
    go to card automatic_sched_stat of stack templates
    doMenu "Copy Card"
    doMenu "New Stack..."
    doMenu "Paste Card"
    go first
    doMenu "Delete Card"
    --fill in  Quarter after ovhl - Ship status - Cumulative Q
    push card
    go to stack ship_data
    put card fld crd_fld_Quart_aft_ovhl into Q_TRANSFER
    if the visible of button active then
      put "active" into STATUS_TRANSFER
    else
      put "inactive" into STATUS_TRANSFER
    end if
    put card fld Cumulative_Quarter + 1 into card fld Cumulative_Quarter
    put card fld Cumulative_Quarter into CUMULATIVE_TRANSFER
```

```
--fill in work center names & MIP for each work center
repeat with i = 1 to the number of cards of bg responsibility_bg
  go to stack ship_data
  go to card 1 of bg responsibility_bg
  put fld WC_resp into VAR1
  put fld MIP_no_resp into VAR2
  pop card
  doMenu "Paste Card"
  put Q_TRANSFER into fld Q_after_ovhl
  put CUMULATIVE_TRANSFER into fld cumul_quarter
  put STATUS_TRANSFER into fld sh_status
  put VAR1 into fld WC_name
  put VAR2 into fld WC_mips
  push card
end repeat
go first
doMenu "Delete Card"
--empty variables to reserve memory
put empty into CUMULATIVE_TRANSFER
put empty into VAR1
put empty into VAR2
-- fill in all the MRC numbers
repeat with i = 1 to the number of cards of bg schedule_page1
  go to card 1 of bg schedule_page1
  repeat with ii = 1 to the number of lines of fld WC_mips
    push card
    put line ii of fld WC_mips into TEMP_MIP
    put TEMP_MIP & "-1" into TEMP_MIP_TO_GO
    go to card TEMP_MIP_TO_GO of stack PMS
    if the result is not empty then
      pop card
      put TEMP_MIP && ": MIP not in the PMS DB" & return
      after fld not_scheduled
    else
      if STATUS_TRANSFER = "active" then
        put fld MRC_of_this_MIP_active into MIP_DATA
      else
        put fld MRC_of_this_MIP_inactive into MIP_DATA
      end if
      pop card
      put MIP_DATA & return after fld WC_mrcs
    end if
  end repeat
  -- clean up empty lines
  repeat with ii = the number of lines of fld WC_mrcs down to 1
    if line ii of fld WC_mrcs is empty then
      delete line ii of fld WC_mrcs
    end if
  end repeat
  -- clean up WC_mrcs field from extra spaces at begining
  repeat with ii = 1 to the number of lines of fld WC_mrcs
    if char 1 of line ii of fld WC_mrcs = " " then
      delete char 1 of line ii of fld WC_mrcs
    end if
  end repeat
end repeat
--empty variables to reserve memory
put empty into TEMP_MIP
put empty into MIP_DATA
--Fill in the info field for all MRCs found
repeat with i = 1 to the number of cards of bg schedule_page1
  go to card 1 of bg schedule_page1
  push card
```

```
  put empty into NOT_FOUND_MRC_VAR
  put empty into INFO_VAR
  repeat with ii = 1 to the number of lines of MRC_VAR
    put line ii of MRC_VAR into INIT_DEST
    put INIT_DEST into DEST
    if char 1 of DEST = "." or char 1 of DEST = "-" then delete char 1 of DEST
    go to card DEST of stack PMS
    if the result is not empty then
      if char 1 of INIT_DEST = "-" or char 1 of INIT_DEST = "." then
        put INIT_DEST && ": MRC not in the PMS DB" & return
        after NOT_FOUND_MRC_VAR
      else
        put " " & INIT_DEST && ": MRC not in the PMS DB" & return
        after NOT_FOUND_MRC_VAR
      end if
    else
      --form status
      put empty into VISIBLE_BUTTONS
      if the visible of button anchor then put "A" after VISIBLE_BUTTONS
      if the visible of button at_sea then put "B" after VISIBLE_BUTTONS
      if the visible of button flag then put "C" after VISIBLE_BUTTONS
      if the visible of button overhaul then put "D" after VISIBLE_BUTTONS
      if the visible of button inactive then put "E" after VISIBLE_BUTTONS
      --form relative maintenance
      put empty into RELATIVES
      repeat with iii = 1 to the number of lines of fld rel_maint
        put line iii of fld rel_maint into TEMP_REL
        if TEMP_REL = "None" then
          put "N" into RELATIVES
        else
          put line iii of fld rel_maint & "*" after RELATIVES
        end if
      end repeat
      --form work centers involved
      put empty into INVDLVED
      repeat with iii = 1 to the number of lines of fld other_wc_involved
        put line iii of fld other_wc_involved into TEMP_INV
        put line iii of fld other_wc_involved & " " after INVOLVED
      end repeat
      put INIT_DEST & "," & line 1 of fld periodicity & "," & fld MH
      & "," & VISIBLE_BUTTONS & "," & RELATIVES & ","
      & INVOLVED & return into TRANSFER
      put TRANSFER after INFO_VAR
    end if
  end repeat
  pop card
  put NOT_FOUND_MRC_VAR after fld not_scheduled
  put INFO_VAR into fld info
end repeat
--empty variables to reserve memory
put empty into MRC_VAR
put empty into NOT_FOUND_MRC_VAR
put empty into INFO_VAR
put empty into INIT_DEST
put empty into DEST
put empty into VISIBLE_BUTTONS
put empty into RELATIVES
put empty into TRANSFER
--do the sorting
if STATUS_TRANSFER = "active" then
  repeat with i = 1 to the number of cards of bg schedule_page1
    go to card 1 of bg schedule_page1
    repeat with ii = 1 to the number of lines of fld info
    if char 1 of TEST_ITEM = "D" then
      put "A" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "2" and char 2 of TEST_ITEM = "D" then
      put "B" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "3" and char 2 of TEST_ITEM = "D" then
      put "C" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "W" then
      put "D" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "2" and char 2 of TEST_ITEM = "W" then
      put "E" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "M" then
      put "F" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "2" and char 2 of TEST_ITEM = "M" then
      put "G" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "O" then
      put "H" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "S" then
      put "I" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 of TEST_ITEM = "A" then
      put "J" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "18" then
      put "K" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "24" then
      put "L" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "30" then
      put "M" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "36" then
      put "N" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "48" then
      put "O" into item 2 of line 11 of fld info
      next repeat
    end if
    if char 1 to 2 of TEST_ITEM = "54" then
      put "P" into item 2 of line 11 of fld info
      next repeat
    end if
```

```
      if char 1 to 2 of TEST_ITEM = "60" then
        put "Q" into item 2 of line 11 of fld info
        next repeat
      end if
      if char 1 of TEST_ITEM = "U" then
        put item 1 of line 11 of fld
        info && ":  Unscheduled Maintenance Job"
        & return after fld not_scheduled
        put empty into line 11 of fld info
        next repeat
      end if
      if char 1 of TEST_ITEM = "R" then
        put item 1 of line 11 of fld
        info && ":  Situation Requirement Job"
        & return after fld not_scheduled
        put empty into line 11 of fld info
        next repeat
      end if
    end repeat
    put fieldsort (fld info,2) into fld info
  end repeat
end if
push card --Automatic Scheduling
--empty variables to reserve memory
put empty into TEST_ITEM
-- CREATE DAs
go to card daily_activ_card of stack templates
doMenu "Copy Card"
pop card
doMenu "Paste Card"
push card --1st DA card in Automatic Stack
push card --1st DA card in Automatic Stack
--Find out how many Work Centers are there
put the number of cards - 1 into DEV_CARDS
--PREPARE NEW STACK's NAME
go to last card of stack planning
put character 1 to 3 of word 1 of fld bg_fld_month_year &&
word 2 of fld bg_fld_month_year into temp2
go prev
go prev
put "DA" && character 1 to 3 of word 1 of fld bg_fld_month_year
& "-" & temp2 into NEW_STACK_NAME
--GET INFO LOOP ----- PREPEARE VARIABLES
repeat with count1 = 1 to 3  --for 3 months
  repeat with BOX_COUNT=1 to 37  --the # of boxes
    put BOX_COUNT into TEMP_BOX_COUNT
    if TEMP_BOX_COUNT > 6 and TEMP_BOX_COUNT < 29 then
      put "bg_fld_date_full_" & BOX_COUNT into WHAT_DAY
      put fld WHAT_DAY into TEMP_DAY
      put TEMP_DAY & return after TEMP_DAY_SHORT_VAR
      convert TEMP_DAY to long date
      put TEMP_DAY & return after LONG_DATE_VAR
      repeat with COUNT_BUTTON = ((BOX_COUNT * 5) - 4) to (BOX_COUNT * 5)
        if the visible of button COUNT_BUTTON is true then
          put (COUNT_BUTTON mod 5) into WHAT_BUTTON
          if WHAT_BUTTON = 0 then
            put 5 into WHAT_BUTTON
          end if
          put WHAT_BUTTON & return after WHAT_BUTTON_VAR
          exit repeat
        end if
      end repeat
    else
      if TEMP_BOX_COUNT > 28 then
```

```
        put (TEMP_BOX_COUNT - 22) into TEMP_BOX_COUNT
      end if
      if the visible of card fld TEMP_BOX_COUNT is false then
        put "bg_fld_date_full_" & BOX_COUNT into WHAT_DAY
        put fld WHAT_DAY into TEMP_DAY
        put TEMP_DAY & return after TEMP_DAY_SNORT_VAR
        convert TEMP_DAY to long date
        put TEMP_DAY & return after LONG_DATE_VAR
        repeat with COUNT_BUTTON = ((BOX_COUNT * 5) - 4) to (BOX_COUNT * 5)
          if the visible of button COUNT_BUTTON is true then
            put (COUNT_BUTTON mod 5) into WHAT_BUTTON
            if WHAT_BUTTON = 0 then
              put 5 into WNAT_BUTTON
            end if
            put WHAT_BUTTON & return after WHAT_BUTTON_VAR
            exit repeat
          end if
        end repeat
      end if
    end if
  end repeat
  go next
end repeat
-- get Work Center names
pop card -- automatic scheduling stack
put the number of cards -1 into NUMBER_OF_WC
put empty into WC_NAMES
repeat with work_centers = 1 to NUMBER_OF_WC
  go to card work_centers
  put fld WC_name & return after WC_NAMES
end repeat
pop card --1st DA card of automatic
push card
--get a copy of first DA card
doMenu "Copy Card"
set name of this stack to NEW_STACK_NAME
push card
put NEW_STACK_NAME into TEMP_NEW_STACK_NAME
delete char 1 to 3 of TEMP_NEW_STACK_NAME
repeat with SET_QUARTER = 1 to the number of cards of bg schedule_page1
  go to card SET_QUARTER of bg schedule_page1
  put TEMP_NEW_STACK_NAME into fld quarter_to_schedule
end repeat
pop card
--empty variables to reserve memory
put empty into DEV_CARDS
put empty into TEMP_NEW_STACK_NAME
put empty into BOX_COUNT
put empty into TEMP_BOX_COUNT
repeat with WORK_CENTERS=1 to NUMBER_OF_WC
  repeat with DAYS_OF_QUARTER=1 to the number of lines of
    LONG_DATE_VAR
    doMenu "Paste Card"
    if RESCN then
      set the highlight of button rescheduled to false
    end if
    put line WORK_CENTERS of WC_NAMES & "-" & line DAYS_OF_QUARTER
    of TEMP_DAY_SHORT_VAR into FORM_CARD_NAME
    set the name of this card to FORM_CARD_NAME
    put line WORK_CENTERS of WC_NAMES into fld bg_fld_wc_DA
    put line DAYS_OF_QUARTER of LONG_DATE_VAR into fld bg_fld_date
    put line DAYS_OF_QUARTER of WNAT_BUTTON_VAR into TEMP_BUT
    show button TEMP_BUT
```

```
        if TEMP_BUT = 2 then put "B" into fld letter_sh_status
        if TEMP_BUT = 3 then put "C" into fld letter_sh_status
        if TEMP_BUT = 4 then put "D" into fld letter_sh_status
        if TEMP_BUT = 5 then put "E" into fld letter_sh_status
        put 0 into fld calculated_mh
        put 0 into fld completed
        -- hide previous arrow on first day of each quarter
        put fld bg_fld_date into TEMP_DATE_TO_CNECK
        convert TEMP_DATE_TO_CHECK to dateItems
        if (item 3 of TEMP_DATE_TO_CNECK = 1) and (item 2
        of TEMP_DATE_TO_CNECK mod 3 = 1) then
          hide button crd_but_go_prev
        end if
      end repeat
      hide button crd_but_go_next
    end repeat
    go to card 1 of bg DA_model
    doMenu "Delete Card"
    --empty variables to reserve memory
    put empty into DAYS_OF_QUARTER
    put empty into LONG_DATE_VAR
    put empty into WORK_CENTERS
    put empty into WC_NAMES
    put empty into TEMP_DAY_SNORT_VAR
    put empty into FORM_CARD_NAME
    put empty into WHAT_BUTTON_VAR
    put empty into TEMP_BUT
    --Actual Scheduling
    go to stack NEW_STACK_NAME
    if STATUS_TRANSFER = "active" then
      send mouseUp to bg button do_it_active
    else
      send mouseUp to bg button do_it_inactive
    end if
    if the freesize of this stack > 0.15 * the size of this stack then
      doMenu "Compact Stack"
    end if
  end if
end mouseUp

***** BACKGROUNO #13: bar_code *****
on newCard
  put the date into fld bg_fld_date
end newCard

 ***** BKGND #13, FIELD #6: FSN *****
on returnInField
  put me into fld bar_coded_fld
  click at 135,211
end returnInField

 ***** BKGND #13, FIELD #9: Out_by *****
on returnInField
  put me into fld bar_coded_fld
  click at 135,211
end returnInField

 ***** BKGND #13, BUTTON #1: print_it *****
on mouseUp
  -- ask the user for the quantity of labels he needs printed
  ask "Now many copies do you need ?" with "1"
  -- if he changed his mind...
```

```
    exit mouseUp
    -- but if not...
  else
    -- store the copies he needs temporarily
    put it into TIMES_TO_PRINT
    -- suspend screen drawing
    lock screen
    -- hide the buttons
    hide me
    hide bg button bg_but_throw_away
    hide bg button print_and_store
    hide bg button print_and_store_top
    -- do_it for all copies
    repeat with i = 1 to TIMES_TO_PRINT
      print this card
    end repeat
    -- throw card away
    doMenu Delete Card
    -- go to driver's menu card
    go prev
    -- restore screen drawing
    unlock screen
  end if
  -- exit event
end mouseUp

 ***** BKGND #13, BUTTON #2: bg_but_throw_away *****
on mouseUp
  lock screen
  doMenu Delete Card
  go prev
  unlock screen
end mouseUp

 ***** BKGND #13, BUTTON #3: print_and_store *****
on mouseUp
  -- ask the user to which stack he wants this card stored
  ask "Store this label to:" with "General Label Storage"
  if it is empty or it is "CANCEL" then
    send mouseUp to bg button print_it
  else
    -- set cursor to wait
    set cursor to 4
    -- suspend screen refreshing
    lock screen
    -- check if stack exists
    put it into DEST
    push card
    doMenu "Copy Card"
    go to card 1 of stack DEST
    if the result is not empty then
      unlock screen
      doMenu "New Stack..."
      lock screen
      doMenu "Paste Card"
      go first
      doMenu "Delete Card"
      pop card
      unlock screen
      send mouseUp to bg button print_it
    else
      doMenu "Paste Card"
      pop card
      unlock screen
```

```
      send mouseUp to bg button print_it
    end if
  end if
  -- restore screen drawing
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #13, BUTTON #4: print_and_store_top *****
on mouseDown
  set the hilite of bg button print_and_store to true
end mouseDown
on mouseUp
  set the hilite of bg button print_and_store to false
  send mouseUp to bg button print_and_store
end mouseUp

 ***** BKGND #21, BUTTON #2: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #21, BUTTON #4: New Button *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #21, BUTTON #5: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #21, BUTTON #6: d_field *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  subtract 5 from it
  if it < "0" then
    put "0" into it
  end if
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #21, BUTTON #7: d_field *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  add 5 to it
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #21, BUTTON #8: u_field *****
```

```
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field ship_system into it
  add 5 to it
  set the lockscreen to true
  set the scroll of field ship_system to it
  set the scroll of field pubs to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #21, BUTTON #9: print this mip *****
on mouseUp
  set cursor to watch
  lock screen
  put field ship_system into ship_temp
  put field pubs into pubs_temp
  put field config into config_temp
  put  field card_date into date_temp
  put field MIP into MIP_temp
  put field card_name into name_temp
  put 0 into GLOBAL_PAGES
  put 0 into GLOBAL_TOTAL_PAGES
  push card
  go next card
  put field test into test_temp
  put field MRC into MRC_temp
  put field MRCdescript into descr_temp
  put field period into period_temp
  put field rates into rates_temp
  put field man into man_temp
  put field relman into relman_temp
  --calculate pages required
  put 0 into FULL_PAGES
  put the number of lines of fld MRCdescript into PROCEDURE_LINES
  put (PROCEDURE_LINES - 27) div 44 + 1 into FULL_PAGES
  put (PROCEDURE_LINES - 27) mod 44 into REMAINING
  if REMAINING > 0 then
    put FULL_PAGES + 1 into FULL_PAGES
  end if
  put FULL_PAGES into GLOBAL_TOTAL_PAGES
  go to stack MIP_printed
  repeat with i=1 to(the number of cards - 1)
    go to card 2
    domenu "DELETE CARD"
  end repea
  go to card "print_page_1" of stack "templates"
  doMenu "NEW CARD"
  doMenu "CUT CARD"
  go to stack MIP_printed
  doMenu "LAST"
  doMenu "PASTE CARD"
  put  ship_temp into field ship_system
  put  pubs_temp into field pubs
  put  config_temp into field config
  put  date_temp into field card_date
  put  MIP_temp into  field MIP
  put  name_temp into field card_name
  put 1 into count
  repeat while count < 3
    put line count of test_temp & return after field test
    put line count of descr_temp & return after field MRCdescript
    put  line count of period_temp  & return after field period
    put line count of rates_temp  & return after field rates
    put line count of man_temp  & return after field man
    put line count of relman_temp & return after field relman
    put count + 1 into count
  end repeat
  go to card "print_page_2" of stack "templates"
  doMenu "NEW CARD"
  doMenu "CUT CARD"
  go to stack MIP_printed
  doMenu "LAST"
  doMenu "PASTE CARD"
  repeat while count < "28"
    put line count of test_temp & return after field test
    put line count of MRC_temp & return after field MRC
    put line count of descr_temp & return after field MRCdescript
    put  line count of period_temp  & return after field period
    put line count of rates_temp  & return after field rates
    put line count of man_temp  & return after field man
    put line count of relman_temp & return after field relman
    put count + 1 into count
  end repeat
  put the date into fld date_printed_pr
  put 1 into GLOBAL_PAGES
  put GLOBAL_PAGES into fld page_no_pr
  put GLOBAL_TOTAL_PAGES into fld page_of_pr
  put  PROCEDURE_LINES - "27" into  PROCEDURE_LINES
  put 47 into c_var
  put 72 into c_var2
  repeat until PROCEDURE_LINES <= 0
    go to card "print_page_3" of stack "templates"
    doMenu "NEW CARD"
    doMenu "CUT CARD"
    go to stack MIP_printed
    doMenu "LAST"
    doMenu "PASTE CARD"
    put  MIP_temp into  field MIP
    repeat while count < c_var
      put line count of test_temp & return after field test
      put line count of MRC_temp & return after field MRC
      put line count of descr_temp & return after field MRCdescript
      put  line count of period_temp  & return after field period
      put line count of rates_temp  & return after field rates
      put line count of man_temp  & return after field man
      put line count of relman_temp & return after field relman
      put count + 1 into count
    end repeat
    put PROCEDURE_LINES - 19 into PROCEDURE_LINES
    go to card "print_page_2" of stack "templates"
    doMenu "NEW CARD"
    doMenu "CUT CARD"
    go to stack MIP_printed
    doMenu "LAST"
    doMenu "PASTE CARD"
    repeat while count < c_var2
      put line count of test_temp & return after field test
      put line count of MRC_temp & return after field MRC
      put line count of descr_temp & return after field MRCdescript
      put  line count of period_temp  & return after field period
      put line count of rates_temp  & return after field rates
      put line count of man_temp  & return after field man
    end repeat
    put PROCEDURE_LINES - "25" into PROCEDURE_LINES
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
    put c_var + "44" into c_var
    put c_var2 + "44" into c_var2
  end repeat
  put the number of cards  into c_total
  go to card 1 of stack MIP_printed
  domenu "DELETE CARD"
  print all cards
  doMenu "Compact Stack"
  pop card
  unlock screen
end mouseUp

 ***** BKGND #21, BUTTON #10: append *****
on mouseUp
  global DEST
  set cursor to 4
  push card
  push card
  go to stack DEST
  put the number of cards into HOW_MANY_CARDS
  pop card
  go last
  repeat with i = 1 to HOW_MANY_CAROS
    put "Processing MIP :" && OEST && "                 card" && i && "out of" &&
    HOW_MANY_CARDS into msg
    push card --last card of PMS stack
    go to card 1 of stack DEST
    if the short name of this bg is "MIP-1st page" then
      go to stack PMS
      go to first card of bg "MIP-1st page"
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "card_date" into VAR1
      put field "pubs" into VAR2
      put field "config" into VAR3
      put field "MIP" into VAR4
      put field "card_NAME" into VAR5
      put field "ship_system" into VAR6
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "card_date"
      put VAR2 into field "pubs"
      put VAR3 into field "config"
      put VAR4 into field "MIP"
      put VAR5 into field "card_NAME"
      put VAR6 into field "ship_system"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "mip 2nd page" then
      go to stack PMS
      go to first card of bg "mip 2nd page"
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "MIP" into VAR1
      put field "test" into VAR2
      put field "MRC" into VAR3
      put field "MRCdescript" into VAR4
      put field "period" into VAR5
      put field "rates" into VAR6
      put field "man" into VAR9
      put field "relman" into VAR10
      put field "card_NAME" into VAR11
      put field "where_am_i" into VAR12
      put field "MRC_of_this_MIP_active" into VAR13
      put field "MRC_of_this_MIP_inactive" into VAR14
      -- put field "data_MRC" into VAR15
      -- put field "in_data_MRC" into VAR16
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MIP"
      put VAR2 into field "test"
      put VAR3 into field "MRC"
      put VAR4 into field "MRCdescript"
      put VAR5 into field "period"
      put VAR6 into field "rates"
      put VAR9 into field "man"
      put VAR10 into field "relman"
      put VAR11 into field "card_NAME"
      put VAR12 into field "where_am_i"
      put VAR13 into field "MRC_of_this_MIP_active"
      put VAR14 into field "MRC_of_this_MIP_inactive"
      -- put VAR15 into field "data_MRC"
      -- put VAR16 into field "in_data_MRC"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "MRC - 1st page." then
      go to stack PMS
      go to first card of bg "MRC - 1st page."
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "MRCno" into VAR1
      put field "what_equipment" into VAR2
      put field "MH" into VAR3
      put field "elapsed" into VAR4
      put field "mrcode" into VAR5
      put field "periodicity" into VAR6
      put field "date_edited" into VAR9
      put field "rel_maint" into VAR11
      put field "mr_description_field" into VAR12
      put field "rates_mh_field" into VAR13
      put field "equipment_field" into VAR14
      put field "subsystem_field" into VAR15
      put field "data" into VAR16
      put field "ship_system_field" into VAR17
      put field "system_field" into VAR18
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MRCno"
      put VAR2 into field "what_equipment"
      put VAR3 into field "MH"
      put VAR4 into field "elapsed"
      put VAR5 into field "mrcode"
      put VAR6 into field "periodicity"
      put VAR9 into field "date_edited"
      put VAR11 into field "rel_maint"
      put VAR12 into field "mr_description_field"
      put VAR13 into field "rates_mh_field"
      put VAR14 into field "equipment_field"
      put VAR15 into field "subsystem_field"
      put VAR16 into field "data"
      put VAR17 into field "ship_system_field"
      put VAR18 into field "system_field"
      set the name of this card to NAME_VAR
      next repeat
    end if
    if the short name of this bg is "MRC 2nd page" then
      go to stack PMS
      go to first card of bg "MRC 2nd page"
      doMenu "Copy Card"
      pop card --last of PMS
      doMenu "Paste Card"
      push card
      go to card 1 of stack DEST
      put field "MRCno" into VAR1
      put field "what_equipment" into VAR2
      put field "procedure" into VAR3
      put the short name of this card into NAME_VAR
      pop card
      put VAR1 into field "MRCno"
      put VAR2 into field "what_equipment"
      put VAR3 into field "procedure"
      set the name of this card to NAME_VAR
      next repeat
    end if
  end repeat
  pop card
end mouseUp

 ***** BKGND #21, BUTTON #11: New Button *****
on mouseUp
  push card
  lock screen
  go next
  go next
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #21, BUTTON #12: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #21, BUTTON #13: New Button *****
on mouseUp
  ask "Which MIP do you wish to view ?"
```

```
    exit mouseup
  else
    lock screen
    push card
    go to card 1
    --put quote & it & quote into it
    put binary (mip_list,it) into it
    if it is not "not found" then
      put item 2 of line it of fld mip_list into it
      go to card it
    else
      pop card
      unlock screen
      answer ("Card not in the PMS database !!!")
    end if
  end if
end mouseUp

 ***** BKGND #22, FIELD #2: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                    "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "      " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #3: MRC *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "                    "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "      " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #4: MRCdescript *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "        " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #5: period *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "        " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #6: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "        " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #7: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "        " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, FIELD #8: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field MRC <> "          "
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  if "        " is in line it of field MRC then
    exit mouseUp
  else
    put line it of field MRC into it
    if char 1 of it = "." then delete char 1 of it
    push card
    go to card it
  end if
end mouseUp

 ***** BKGND #22, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #22, BUTTON #3: New Button *****
on mouseUp
  go previous
  send mouseup to  bg button "print this mip"
end mouseUp

 ***** BKGND #22, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #22, BUTTON #14: New Button *****
on mouseUp
  visual effect scroll right
  pop card
end mouseUp

 ***** BKGND #22, BUTTON #16 *****
on mouseUp
  set the visible of fld MRC_of_this_MIP_active to not the visible of
  fld MRC_of_this_MIP_active
  set the visible of fld MRC_of_this_MIP_inactive to the visible of
  fld MRC_of_this_MIP_active
end mouseUp

 ***** BKGND #22, BUTTON #17: New Button *****
on mouseUp
  ask "Which MIP do you wish to view ?"
  if it is empty or it is "Cancel" then
    exit mouseup
  else
    lock screen
    push card
    go to card 1
    --put quote & it & quote into it
    put binary (mip_list,it) into it
    if it is not "not found" then
      put item 2 of line it of fld mip_list into it
      go to card it
    else
      pop card
      unlock screen
      answer ("Card not in the PMS database !!!")
    end if
  end if
end mouseUp

 ***** BKGND #22, BUTTON #18: New Button *****
on mouseUp
  push card
  lock screen
  go next
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #22, BUTTON #19: down *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field relman into it
  add 156 to it
  set the lockscreen to true
  set the scroll of field test to it
  set the scroll of field MRC to it
  set the scroll of field MRCdescript to it
  set the scroll of field period to it
  set the scroll of field rates to it
  set the scroll of field man to it
  set the scroll of field relman to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #22, BUTTON #20: left *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #22, BUTTON #21: go_top *****
on mouseDown
  set the lockscreen to true
  set the scroll of field test to 0
```

```
  set the scroll of field MRCdescript to 0
  set the scroll of field period to 0
  set the scroll of field rates to 0
  set the scroll of field man to 0
  set the scroll of field relman to 0
  set the lockscreen to false
end mouseDown

 ***** BKGND #22, BUTTON #22: up *****
on mousestillDown
  send mouseDown to me
end mousestillDown

on mouseDown
  put the scroll of field relman into it
  subtract 156 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of field test to it
  set the scroll of field MRC to it
  set the scroll of field MRCdescript to it
  set the scroll of field period to it
  set the scroll of field rates to it
  set the scroll of field man to it
  set the scroll of field relman to it
  set the lockscreen to false
end mouseDown

***** BACKGROUND #23: MRC - 1st page. *****
on do_select
  put "Add,Normal Duty,At Sea,Noliday,Overhaul,Inactive Status" into menu1
  put return & "Remove,Normal Duty,At Sea,Holiday,Overhaul,Inactive Status,This" after me
nu1
  get HPopupMenu(menu1,0,the mouseV,the mouseH)
  put 0 into countvisible
  put TRUE into CHECK
  repeat with i = 1 to 5
    if the visible of button i then put countvisible + 1 into countvisible
  end repeat
  if countvisible = 1 then put FALSE into CNECK
  if it is not zero then
    put item 1 of it into theline
    put item 2 of it into theItem
    if theline = 1 and theItem = 2 then
      show button anchor
    end if
    if theline = 1 and theItem = 3 then
      show button at_sea
    end if
    if theline = 1 and theItem = 4 then
      show button flag
    end if
    if theline = 1 and theItem = 5 then
      show button overhaul
    end if
    if theline = 1 and theItem = 6 then
      show button inactive
    end if
    if theline = 2 and theItem = 2 and CHECK then
      hide button anchor
    end if
    if theline = 2 and theItem = 3 and CNECK then
      hide button at_sea
    if theline = 2 and theItem = 4 and CHECK then
      hide button flag
    end if
    if theline = 2 and theItem = 5 and CHECK then
      hide button overhaul
    end if
    if theline = 2 and theItem = 6 and CHECK then
      hide button inactive
    end if
    if theline = 2 and theItem = 7 and CHECK then
      hide the target
    end if
    adjust
  end if
end do_select

on adjust
  put -5 into newloc
  repeat with i = 1 to 5
    if the visible of button i then
      put newloc + 36 into newloc
      set the loc of button i to newloc,39
    end if
  end repeat
end adjust

on enterkey
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  if SEARCH_STRING_VAR is not empty then
    --go next
    if SEARCH_FIELD_VAR = "global" then
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    else
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD_VAR
      if the result is not empty then
        answer "String not found..."
      end if
    end if
  end if
end enterkey

 ***** BKGND #23, BUTTON #1: print_mrc *****
on mouseUp
  set cursor to 4
  lock screen
  put 0 into GLOBAL_PAGES
  put 0 into GLOBAL_TOTAL_PAGES
  --clear destination
  put the id of this card into ID_COME_BACK
  push card
  go to stack MRC_print
  repeat with i = 1 to (the number of cards - 2)
    go to card 3
    doMenu "Delete Card"
  end repeat
  pop card      --Stack:empty
  put item 1 of line 4 of fld data + 1 into NUMBER_OF_PROCEDURE_PAGES
  put item 2 of line 4 of fld data into DIAGRAMS
  put item 3 of line 4 of fld data into TABLES
  --make second page of pms
```

```
  push card  --Stack:click card,1st mrc card
  put empty into TRANSFER_PROCEDURE
  repeat with PROC_PAGES = 1 to NUMBER_OF_PROCEDURE_PAGES
    go next --2nd mrc card
    put fld procedure after TRANSFER_PROCEDURE
  end repeat
  repeat with PROC_LINES = 1 to the number of lines of TRANSFER_PROCEDURE
    if line PROC_LINES of TRANSFER_PROCEDURE ="---------------------------------------------
---------------------------------------------" then
      put "____________________________________________________________________________" i
nto
      line PROC_LINES of TRANSFER_PROCEDURE
    end if
  end repeat
  --calculate & make pages required
  put 0 into FULL_PAGES
  put the number of lines of TRANSFER_PROCEDURE into PROCEDURE_LINES
  if PROCEDURE_LINES > 25 then
    put ((PROCEDURE_LINES - 25) mod 47) into REMAINING_LINES
    put ((PROCEDURE_LINES - 25) div 47) into FULL_PAGES
    if REMAINING_LINES > 0 then put FULL_PAGES + 1 into FULL_PAGES
  end if
  repeat with i = 1 to FULL_PAGES
    go to card print_mrc_page3 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page2
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  repeat with i = 1 to DIAGRAMS
    go to card print_mrc_page4 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page7
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  put TABLES div 2 into TABLE_PAIRS
  put TABLES mod 2 into TABLE_REMAIN
  repeat with i = 1 to TABLE_PAIRS
    go to card print_mrc_page4 of stack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card    --Stack:click card,1st mrc card
    go to card print_mrc_page5
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  repeat with i = 1 to TABLE_REMAIN
    go to card print_mrc_page4 of atack Templates
    doMenu "Copy Card"
    push card   --Stack:click card,1st mrc card,template
    go to last card of stack MRC_print
    doMenu "Paste Card"
    pop card  --Stack:click card,1st mrc card
    go to card print_mrc_page6
    doMenu "Copy Card"
    go to last card of stack MRC_print
    doMenu "Paste Card"
  end repeat
  put (the number of cards / 21) into GLOBAL_TOTAL_PAGES
  pop card  --Stack:click card.  We are back to first MRC
  --make page 1
  put empty into WHAT_STATUS
  if the visible of button anchor then
    put "anchor," after WHAT_STATUS
  end if
  if the visible of button at_sea then put "at_sea," after WHAT_STATUS
  if the visible of button flag then put "flag," after WHAT_STATUS
  if the visible of button overhaul then put "overhaul," after WHAT_STATUS
  if the visible of button inactive then put "inactive," after WHAT_STATUS
  put fld what_equipment into VAR1
  put fld MRCno into VAR2
  put fld ship_system_field into VAR3
  put fld subsystem_field into VAR4
  put fld mrcode into VAR5
  put fld periodicity into VAR6
  put fld system_field into VAR7
  put fld equipment_field into VAR8
  put fld rates_mh_field into VAR9
  put fld mr_description_field into VAR10
  put fld MH into VAR11
  put fld elapsed into VAR12
  put fld date_edited into VAR13
  push card   --Stack:click card,1st of mrc
  go to card 1 of stack MRC_print
  hide bg button anchor
  hide bg button at_sea
  hide bg button flag
  hide bg button overhaul
  hide bg button inactive
  if WHAT_STATUS contains "anchor" then show bg button anchor
  if WHAT_STATUS contains "at_sea" then show bg button at_sea
  if WHAT_STATUS contains "flag" then show bg button flag
  if WHAT_STATUS contains "overhaul" then show bg button overhaul
  if WHAT_STATUS contains "inactive" then show bg button inactive
  put -5 into newloc
  if the visible of bg button anchor then
    put newloc + 36 into newloc
    set the loc of bg button anchor to newloc,39
  end if
  if the visible of bg button at_sea then
    put newloc + 36 into newloc
    set the loc of bg button at_sea to newloc,39
  end if
  if the visible of bg button flag then
    put newloc + 36 into newloc
    set the loc of bg button flag to newloc,39
  end if
  if the visible of bg button overhaul then
    put newloc + 36 into newloc
    set the loc of bg button overhaul to newloc,39
  end if
  if the visible of bg button inactive then
    put newloc + 36 into newloc
```

```
  end if
  put VAR1 into fld what_equipment_pr
  put VAR2 into fld MRCno_pr
  put VAR3 into fld ship_system_field_pr
  put VAR4 into fld subsystem_field_pr
  put VAR5 into fld mrcode_pr
  put VAR6 into fld periodicity_pr
  put VAR7 into fld system_field_pr
  put VAR8 into fld equipment_fld_pr
  put VAR9 into fld rates_mh_field_pr
  put VAR10 into fld mr_description_field_pr
  put VAR11 into fld MH_pr
  put VAR12 into fld elapsed_pr
  put VAR13 into fld date_edited_pr
  go to card 2 of stack MRC_print
  put 0 into START_LINE
  put 0 into LAST_LINE
  put LAST_LINE + 1 into START_LINE
  put START_LINE + 24 into LAST_LINE
  put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
  fld procedure_pr
  put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
  put the date into fld date_printed_pr
  put 1 into GLOBAL_PAGES
  put GLOBAL_PAGES into fld page_no_pr
  put GLOBAL_TOTAL_PAGES into fld page_of_pr
  put the id of this card into TEMPID
  repeat until PROCEDURE_LINES <= 0
    go next
    put VAR2 into fld MRCno_pr
    put LAST_LINE + 1 into START_LINE
    put START_LINE + 21 into LAST_LINE
    put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
    fld procedure_pr
    put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
    go next
    put the id of this card into TEMPID
    put LAST_LINE + 1 into START_LINE
    put START_LINE + 24 into LAST_LINE
    put line START_LINE to LAST_LINE of TRANSFER_PROCEDURE into
    fld procedure_pr
    put PROCEDURE_LINES - (LAST_LINE - START_LINE + 1) into PROCEDURE_LINES
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
  end repeat
  pop card   --Stack:click card.  We are at 1st mrc
  repeat for NUMBER_OF_PROCEDURE_PAGES
    go next
  end repeat
  repeat with 1 = 1 to DIAGRAMS
    go next   --diagram card
    push card  --Stack:click card,diagram card
    put card field ledgent into TRANSFER_LEDGENT
    choose select tool
    drag from 23,74 to 481,281
    type "c" with commandkey
    go to stack MRC_print
    go to TEMPID
    go next
    type "v" with commandkey
    drag from 107,188 to 107,208
    put VAR2 into fld MRCno_pr
    go next
    put the id of this card into TEMPID
    put TRANSFER_LEDGENT into fld ledgent_pr
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
    pop card     --Stack:click card.  We are at diagram card
  end repeat
  repeat with i = 1 to TABLES
    go next   --table card
    push card  --Stack:click card,table card
    choose select tool
    drag from 23,74 to 481,281
    type "c" with commandkey
    go to stack MRC_print
    go to TEMPIO
    go next
    put the id of this card into TEMPID
    type "v" with commandkey
    if i mod 2 = 0 then
      drag from 107,188 to 107,156
      put the date into fld date_printed_pr
      put GLOBAL_PAGES + 1 into GLOBAL_PAGES
      put GLOBAL_PAGES into fld page_no_pr
      put GLOBAL_TOTAL_PAGES into fld page_of_pr
    else
      drag from 107,188 to 107,208
      choose browse tool
      put VAR2 into fld MRCno_pr
    end if
    pop card     --Stack:click card.  We are at table card
  end repeat
  go to stack MRC_print
  go last
  if the short name of this card = "print_mrc_page6" then
    put the date into fld date_printed_pr
    put GLOBAL_PAGES + 1 into GLOBAL_PAGES
    put GLOBAL_PAGES into fld page_no_pr
    put GLOBAL_TOTAL_PAGES into fld page_of_pr
  end if
  doMenu "Compact Stack"
  go first
  print all cards
  go to stack "PMS"
  go to ID_COME_BACK
  unlock screen
end mouseUp

 ***** BKGND #23, BUTTON #2: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #23, BUTTON #3: to argos *****
on mouseUp
  push card
  lock screen
  if line 5 of fld data = "ARGOS" then
    pop card
    unlock screen
    answer "No conection to ARGOS has been established..."
  else
    put line 5 of fld data into it
    go to stack ARGOS
    go to card it
    unlock screen
  end if
end mouseUp

 ***** BKGND #23, BUTTON #4: New Button *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #23, BUTTON #5: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #23, BUTTON #6: New Button *****
on mouseUp
  put line 1 of fld data into it
  delete char 1 to 4 of it
  push card
  go to card it
end mouseUp

 ***** BKGND #23, BUTTON #7: New Button *****
on mouseUp
  push card
  lock screen
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #23, BUTTON #8: open repair parts *****
on mouseUp
  lock screen
  show bg button "repair parts"
  show bg button close
  show card field "repair parts field"
  unlock screen
end mouseUp

 ***** BKGND #23, BUTTON #9: New Button *****
on mouseUp
  push card
  -- This part between the asterisks sould be removed when real pictures
  -- will be entered to the system. Now a couple of demonstration pictures
  -- are provided as space holders in order to demonstrate the
  -- capability of the system.  The pictures were taken by the use of an
  -- ordinary 35 mm camera. The part of code after the  asterisks
  -- should be inserted here, and the "--" should be removed from the
  -- begining of the lines.
  visual effect dissolve
  go to stack movies
    -- repeat until fld page_no = 1
  --    go prev
  -- end repeat
  -- if line 6 of fld data = "MOVIE" then
  --    pop card
  --    unlock screen
  --    answer "No pictures available for this job..."
  -- else
  --    visual effect dissolve
  --    put field "MRCno" & "-movie" into it
  --    push card
  --    go to card it of stack movies
  -- end if
end mouseUp

 ***** BKGND #23, BUTTON #11: close *****
on mouseUp
  lock screen
  hide card field "repair parts field"
  hide me
  hide bg button "Repair Parts"
  unlock screen
end mouseUp

 ***** BKGND #23, BUTTON #13: locker *****
on mouseDown
  put "Unlock Description" into menu1
  get HPopupMenu(menu1,0,193,298)
end mouseDown

on mouseUp
  hide me
  show bg button unlocker
  set the locktext of fld mr_description_field to false
end mouseUp

 ***** BKGND #23, BUTTON #16: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #23, BUTTON #17: forget_it *****
on mouseDown
  put return & "Exit Search" into menu
  get HPopupMenu (menu,0,125,335)
end mouseDown
on mouseUp
  lock screen
  show bg button search
  hide bg button "search for:"
  hide bg button search_menu
  hide bg button search_win
  hide bg button search_now
  hide fld search_by
  hide fld search_string
  hide me
  put empty into fld search_string
  unlock screen
end mouseUp

 ***** BKGND #23, BUTTON #18: search_menu *****
on mouseDown
  global SEARCH_FIELD_VAR
  global DISPLAY
  put "MRC #" into menu
```

```
  put return & "Subsystem" after menu
  put return & "System" after menu
  put return & "Equipment" after menu
  put return & "MRC Code" after menu
  put return & "Periodicity" after menu
  put return & "Rates" after menu
  put return & "Total mh" after menu
  put return & "Description" after menu
  get HPopupMenu (menu,0,the mouseV - 40,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    put line theline of menu into fld search_by
    if theline = 1 then
      put "MRCno" into SEARCH_FIELD_VAR
    end if
    if theline = 2 then
      put "ship_system_field" into SEARCH_FIELD_VAR
    end if
    if theline = 3 then
      put "subsystem_field" into SEARCH_FIELD_VAR
    end if
    if theline = 4 then
      put "system_field" into SEARCH_FIELD_VAR
    end if
    if theline = 5 then
      put "equipment_field" into SEARCH_FIELD_VAR
    end if
    if theline = 6 then
      put "mrcode" into SEARCH_FIELD_VAR
    end if
    if theline = 7 then
      put "periodicity" into SEARCH_FIELD_VAR
    end if
    if theline = 8 then
      put "rates_mh_field" into SEARCH_FIELD_VAR
    end if
    if theline = 9 then
      put "mh" into SEARCH_FIELD_VAR
    end if
    if theline = 10 then
      put "mr_description_field" into SEARCH_FIELD_VAR
    end if
    put line theline of menu into DISPLAY
  else
    put "Global Search" into fld search_by
    put "global" into SEARCH_FIELD_VAR
  end if
end mouseDown
on mouseUp
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #23, BUTTON #19: search_now *****
on mouseDown
  put return & "Find it" into menu
  get HPopupMenu (menu,0,125,335)
end mouseDown

on mouseUp
  global SEARCH_FIELD_VAR
  global SEARCH_STRING_VAR
  push card
  set cursor to 4
  put fld search_string into SEARCH_STRING_VAR
  put length(SEARCH_STRING_VAR) into LAST_CHAR
  if char LAST_CHAR of SEARCH_STRING_VAR = return then
    put empty into char LAST_CHAR of SEARCH_STRING_VAR
  end if
  put empty into fld search_string
  send mouseUp to bg button forget_it
  if SEARCH_STRING_VAR is not empty then
    --go next
    if SEARCH_FIELD_VAR = "global" then
      find chars SEARCH_STRING_VAR
      if the result is not empty then
        pop card
        beep
        answer "String not found..."
      end if
    else
      find chars SEARCH_STRING_VAR in fld SEARCH_FIELD VAR
      if the result is not empty then
        pop card
        beep
        answer "String not found..."
      end if
    end if
  end if
  unlock screen
end mouseUp

 ***** BKGND #23, BUTTON #21: search *****
on mouseUp
  global SEARCH_FIELD_VAR
  put "global" into SEARCH_FIELD_VAR
  lock screen
  put "Global Search" into fld search_by
  put empty into fld search_string
  show bg button search_win
  show bg button search_menu
  show bg button search_now
  show bg button forget_it
  show bg button "search for:"
  show fld search_by
  show fld search_string
  hide me
  unlock screen
  click at the loc of fld search_string
end mouseUp

 ***** BKGND #23, BUTTON #22: dt_opened *****
on mouseUp
  hide me
  show bg button dt_closed
  set the visible of fld data to not the visible of fld data
  set the visible of fld "rel_maint" to the visible of fld data
  set the visible of fld "other_wc_involved" to the visible of fld data
end mouseUp

 ***** BKGND #23, BUTTON #23: dt_closed *****
on mouseUp
  hide me
  show bg button dt_opened
  set the visible of fld data to not the visible of fld data
  set the visible of fld "rel_maint" to the visible of fld data
  set the visible of fld "other_wc_involved" to the visible of fld data
end mouseUp
```

```
***** BACKGROUND #24: MRC 2nd page *****
on opencard
  if fld page_no = fld of pages then hide button go_next
  else
    show button go_next
  end if
end opencard

 ***** BKGND #24, BUTTON #2: Libraries *****
on mouseUp
  push card
  go to stack "TIM-3"
end mouseUp

 ***** BKGND #24, BUTTON #3: New Button *****
on mouseUp
  push card
  repeat until fld page_no = 1
    go prev
  end repeat
  send mouseUp to bg button print_mrc
  pop card
end mouseUp

 ***** BKGND #24, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #24, BUTTON #5: to argos *****
on mouseUp
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  if line 5 of fld data = "ARGOS" then
    pop card
    unlock screen
    answer "No conection to ARGOS has been established..."
  else
    put line 5 of fld data into it
    go to stack ARGOS
    go to card it
    unlock screen
  end if
end mouseUp

 ***** BKGND #24, BUTTON #6: New Button *****
on mouseUp
  go prev
end mouseUp

 ***** BKGND #24, BUTTON #7: New Button *****
on mouseUp
  visual effect dissolve
  put field "MRC-no" & "-movie" into it
  push card
  go to card it
end mouseUp

 ***** BKGND #24, BUTTON #8: New Button *****
on mouseUp
```

```
  pop card
end mouseUp

 ***** BKGND #24, BUTTON #9: New Button *****
on mouseUp
  put field "mip to go" into it
  push card
  go to card it
end mouseUp

 ***** BKGNO #24, BUTTON #10: New Button *****
on mouseUp
  push card
  lock screen
  repeat until fld page_no = 1
    go prev
  end repeat
  get line 2 of field data
  go to card it of stack Log
  if the result is not empty then
    pop card
    answer "No Log card for this piece of equipment"
    unlock screen
  else
    unlock screen
  end if
end mouseUp

 ***** BKGND #24, BUTTON #11: New Button *****
on mouseUp
  pop card
end mouseUp

 ***** BKGND #24, BUTTON #12: New Button *****
on mouseUp
  ask "WHICN MRC DO YOU WISN TO VIEW" with "MRC NUMBER"
  if (it is empty)  then
    exit mouseup
  else
    lock screen
    find whole it in field "MRC"
    if the result = empty then
      put the foundtext into dest
      push card
      go to card dest
      unlock screen
    else
      unlock screen
      exit mouseup
    end if
  end if
end mouseUp

***** CARD #7, FIELD #1: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup
```

```
-- This function togles the two different views of the days - manhours
-- destribution.  It shows the list mode
on mouseUp
  -- show the list
  show card fld days_mh
  -- change the button to the inverse one
  show button show_chart
  -- hide this button
  hide me
  -- exit event
end mouseUp

***** CARD #8, BUTTON #2: show_chart *****
-- This function togles the two different views of the days - manhours
-- destribution.  It shows the graph mode
on mouseUp
  -- hide the list so that the graph under it can be seen
  hide card fld days_mh
  -- change the button to the inverse one
  show button show_list
  -- hide this button
  hide me
  -- exit event
end mouseUp

***** CARO #9, BUTTON #8: crd_but_go_prev *****
on mouseUp
  go prev
end mouseUp

***** CARD #9, BUTTON #9: crd_but_go_next *****
on mouseUp
  go next
end mouseUp

***** CARO #10, BUTTON #1: Enter today's date *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARO #10, BUTTON #4: Go to pub *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #5: Modify pubs *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #7: Go to drawing *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #8: Modify drawings *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #10: Dial mfctr *****
on mouseUp
  set the hilite of me to not the hilite of me
```

```
***** CARD #10, BUTTON #11: Remove date *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #12: Enter other date *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #14: Add data *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #15: Modify remarks *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #18: Done *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Cancel check" and
    not the hilite of button "Cancel PMS job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Check job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #19: Cancel check *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Done" and
    not the hilite of button "Cancel PMS job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Check job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #20: Cancel PMS job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Cancel check" and
    not the hilite of button "Done" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Check job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #23: Add a job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Arrange repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Delete a job" and
    not the hilite of button "Transfer a job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Arrange repair job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #24: Delete a job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Arrange repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add a job" and
    not the hilite of button "Transfer a job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Arrange repair job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #25: Transfer a job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Arrange repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Delete a job" and
    not the hilite of button "Add a job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Arrange repair job" to false
    end if
  end if
end mouseUp

***** CARD #10. BUTTON #27: Fill *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked. check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Notes" to true
  else
    -- but if it is checked. uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "See" and
    not the hilite of button "See all" then
      -- and if not. uncheck the the general button too
      set the hilite of button "Notes" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #28: See *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked. check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Notes" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Fill" and
    not the hilite of button "See all" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Notes" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #29: See all *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked. check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Notes" to true
  else
    -- but if it is checked, uncheck it
    -- check if any other button of this category is checked...
    if not the hilite of button "See" and
    not the hilite of button "Fill" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Notes" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #31: Go to log *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked. check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Equipment log DA" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add job to log" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Equipment log DA" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #32: Add job to log *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Equipment log DA" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Go to log" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Equipment log DA" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #34: Done repair *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Cancel check repair" and
    not the hilite of button "Cancel repair job" then
      -- and if not, uncheck the the general button too
  end if
end mouseUp

***** CARD #10, BUTTON #35: Cancel check repair *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Cancel repair job" and
    not the hilite of button "Done repair" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Check repair job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #36: Cancel repair job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Check repair job" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Cancel check repair" and
    not the hilite of button "Done repair" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Check repair job" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #38: Assign job *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Personel" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Rmv person" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Personel" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #39: Rmv person *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Personel" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Assign job" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Personel" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #40: MRC card print *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Print" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Today's DA card - 1 WC" and
    not the hilite of button "Today's DA cards - all" and
    not the hilite of button "MIP card print" and
    not the hilite of button "Log card print" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Print" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #41: Comment *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #42: Modify mfctr data *****
on mouseUp
  set the hilite of me to not the hilite of me
end mouseUp

***** CARD #10, BUTTON #45: EM01 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM06" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-1" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #46: EM06 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM01" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-1" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #48: EM02 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-2" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM03" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-2" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #49: EM03 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-2" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM02" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-2" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #51: EM04 *****
on mouseUp
```

```
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM05" and
    not the hilite of button "ER09" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #52: EM05 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM04" and
    not the hilite of button "ER09" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #53: ER09 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "E-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "EM05" and
    not the hilite of button "EM04" then
      -- and if not, uncheck the the general button too
      set the hilite of button "E-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #56: SS02 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    set the hilite of button "S-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "S-1" to false
  end if
end mouseUp

***** CARD #10, BUTTON #58: SS06 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-2" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "S-2" to false
  end if
end mouseUp

***** CARD #10, BUTTON #60: SS03 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "SS05" and
    not the hilite of button "SS07" then
      -- and if not, uncheck the the general button too
      set the hilite of button "S-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #61: SS05 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "SS03" and
    not the hilite of button "SS07" then
      -- and if not, uncheck the the general button too
      set the hilite of button "S-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #62: SS07 *****
```

```
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "SS05" and
    not the hilite of button "SS03" then
      -- and if not, uncheck the the general button too
      set the hilite of button "S-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #64: SS01 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-4" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "SS04" then
      -- and if not, uncheck the the general button too
      set the hilite of button "S-4" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #65: SS04 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "S-4" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "SS01" then
      -- and if not, uncheck the the general button too
      set the hilite of button "S-4" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #68: OC01 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "SC-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "SC-1" to false
  end if
end mouseUp

***** CARD #10, BUTTON #70: OC02 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "SC-2" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "SC-2" to false
  end if
end mouseUp

***** CARD #10, BUTTON #72: OC03 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "SC-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "SC-3" to false
  end if
end mouseUp

***** CARD #10, BUTTON #75: CS01 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS02" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-1" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #76: CS02 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
```

```
    set the hilite of button "CS-1" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS01" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-1" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #78: CS03 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-2" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "CS-2" to false
  end if
end mouseUp

***** CARD #10, BUTTON #80: CS04 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS05" and
    not the hilite of button "CS06" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #81: CS05 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS04" and
    not the hilite of button "CS06" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #82: CS06 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-3" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS04" and
    not the hilite of button "CS05" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-3" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #84: CS07 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-4" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS08" and
    not the hilite of button "CS09" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-4" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #85: CS08 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-4" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS07" and
    not the hilite of button "CS09" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-4" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #86: CS09 *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "CS-4" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "CS08" and
    not the hilite of button "CS07" then
      -- and if not, uncheck the the general button too
      set the hilite of button "CS-4" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #89: Convert MIP-MRC file to stack *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add more MIP cards" and
    not the hilite of button "MIP-MRC cards in PMS database" and
    not the hilite of button "Delete a MIP set" and
    not the hilite of button "WC responsibility" and
    not the hilite of button "Empty PMS database" then
      -- and if not, uncheck the the general button too
      set the hilite of button Toolbox to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #90: Empty PMS database *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add more MIP cards" and
    not the hilite of button "Convert MIP-MRC file to stack" and
    not the hilite of button "Delete a MIP set" and
    not the hilite of button "WC responsibility" and
    not the hilite of button "MIP-MRC cards in PMS database" then
      -- and if not, uncheck the the general button too
      set the hilite of button Toolbox to false
    end if
  end if
end mouseUp
```

```
***** CARD #10, BUTTON #91: MIP-MRC cards in PMS database *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add more MIP cards" and
    not the hilite of button "Convert MIP-MRC file to stack" and
    not the hilite of button "Delete a MIP set" and
    not the hilite of button "WC responsibility" and
    not the hilite of button "Empty PMS database" then
      -- and if not, uncheck the the general button too
      set the hilite of button Toolbox to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #92: Delete a MIP set *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add more MIP cards" and
    not the hilite of button "Convert MIP-MRC file to stack" and
    not the hilite of button "MIP-MRC cards in PMS database" and
    not the hilite of button "WC responsibility" and
    not the hilite of button "Empty PMS database" then
      -- and if not, uncheck the the general button too
      set the hilite of button Toolbox to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #95: Add new member *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Password to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Remove a member" and
    not the hilite of button "Change member's access" and
    not the hilite of button "Change my password" then
      -- and if not, uncheck the the general button too
```

```
      end if
    end if
end mouseUp

***** CARD #10, BUTTON #96: Remove a member *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Password to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add new member" and
    not the hilite of button "Change member's access" and
    not the hilite of button "Change my password" then
      -- and if not, uncheck the the general button too
      set the hilite of button Password to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #97: Change member's access *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Password to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add new member" and
    not the hilite of button "Remove a member" and
    not the hilite of button "Change my password" then
      -- and if not, uncheck the the general button too
      set the hilite of button Password to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #99: For entire ship *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Spot checks" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "For one WC" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Spot checks" to false
    end if
  end if
```

```
***** CARD #10, BUTTON #100: For one WC *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Spot checks" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "For entire ship" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Spot checks" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #102: Today's DA cards - all *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Print" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Today's DA card - 1 WC" and
    not the hilite of button "MRC card print" and
    not the hilite of button "MIP card print" and
    not the hilite of button "Log card print" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Print" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #103: Today's DA card - 1 WC *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Print" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Today'a DA cards - all" and
    not the hilite of button "MRC card print" and
    not the hilite of button "MIP card print" and
    not the hilite of button "Log card print" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Print" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #104: MIP card print *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Print" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Today's DA card - 1 WC" and
    not the hilite of button "Today's DA cards - all" and
    not the hilite of button "MRC card print" and
    not the hilite of button "Log card print" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Print" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #105: MRC card *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Go DA" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "MIP card" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Go DA" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #106: Log card print *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Print" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Today's DA card - 1 WC" and
    not the hilite of button "Today's DA cards - all" and
    not the hilite of button "MRC card print" and
    not the hilite of button "MIP card print" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Print" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #108: % done diagram - 1 WC *****
```

```
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Reports & messages" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "% done diagram - all WC" and
    not the hilite of button "Bar coded label" and
    not the hilite of button "Navy message" and
    not the hilite of button "Ext. rep req"  then
      -- and if not, uncheck the the general button too
      set the hilite of button "Reports & messages" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #109: % done diagram - all WC *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Reports & messages" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "% done diagram - 1 WC" and
    not the hilite of button "Bar coded label" and
    not the hilite of button "Navy message" and
    not the hilite of button "Ext. rep req"  then
      -- and if not, uncheck the the general button too
      set the hilite of button "Reports & messages" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #110: Navy message *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Reports & messages" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "% done diagram - all WC" and
    not the hilite of button "option 1" and
    not the hilite of button "% done diagram - 1 WC" and
    not the hilite of button "Ext. rep req"  then
      -- and if not, uncheck the the general button too
      set the hilite of button "Reports & messages" to false
    end if
  end if
```

```
***** CARD #10, BUTTON #111: Ext. rep req *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Reports & messages" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "% done diagram - all WC" and
    not the hilite of button "option 1" and
    not the hilite of button "Navy message" and
    not the hilite of button "% done diagram - 1 WC"  then
      -- and if not, uncheck the the general button too
      set the hilite of button "Reports & messages" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #112: MIP card *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Go DA" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "MRC card" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Go DA" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #116: Next quarter *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Schedule" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    set the hilite of button "Schedule" to false
  end if
end mouseUp

***** CARD #10, BUTTON #117: Go *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    -- but if it is checked, uncheck it if no subcategory is checked
    if not the hilite of button "MRC card go" and
    not the hilite of button "MIP card go" and
    not the hilite of button "Log card go" then
      set the hilite of me to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #118: MRC card go *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Go" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "MIP card go" and
    not the hilite of button "Log card go" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Go" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #119: MIP card go *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Go" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "MRC card go" and
    not the hilite of button "Log card go" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Go" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #120: Bar coded label *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Reports & messages" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "% done diagram - all WC" and
    not the hilite of button "% done diagram - 1 WC" and
    not the hilite of button "Navy message" and
```

```
    not the hilite of button "Ext. rep req" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Reports & messages" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #121: Log card go *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Go" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "MIP card go" and
    not the hilite of button "MRC card go" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Go" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #123: Add entry *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Equipment log" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Del entry" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Equipment log" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #125: Change my password *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Password to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add new member" and
    not the hilite of button "Remove a member" and
    not the hilite of button "Change member's access" then
      -- and if not, uncheck the the general button too
      set the hilite of button Password to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #126: Add more MIP cards *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Convert MIP-MRC file to stack" and
    not the hilite of button "MIP-MRC cards in PMS database" and
    not the hilite of button "Delete a MIP set" and
    not the hilite of button "WC responsibility" and
    not the hilite of button "Empty PMS database" then
      -- and if not, uncheck the the general button too
      set the hilite of button Toolbox to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #127: Del entry *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button "Equipment log" to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add entry" then
      -- and if not, uncheck the the general button too
      set the hilite of button "Equipment log" to false
    end if
  end if
end mouseUp

***** CARD #10, BUTTON #128: WC responsibility *****
on mouseUp
  -- find if this option is not checked
  if not the hilite of me then
    -- if it is not checked, check it
    set the hilite of me to true
    -- check the general button too
    set the hilite of button Toolbox to true
  else
    -- but if it is checked, uncheck it
    set the hilite of me to false
    -- check if any other button of this category is checked...
    if not the hilite of button "Add more MIP cards" and
    not the hilite of button "Convert MIP-MRC file to stack" and
    not the hilite of button "MIP-MRC cards in PMS database" and
    not the hilite of button "Empty PMS database" and
    not the hilite of button "Delete a MIP set" then
```

```
      end if
    end if
end mouseUp

***** CARD #12, BUTTON #1: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #2: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #3: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #4: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #5: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #6: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #7: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #8: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #9: New Button *****
on mouseUp
```

```
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #10: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #11: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #12: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #13: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #14: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #15: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #16: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #17: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #18: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARU #12, BUTTON #19: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #20: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #21: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #22: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #23: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #24: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #25: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #26: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #27: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp
```

```
***** CARD #12, BUTTON #28: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #29: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #30: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #31: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #32: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #33: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #34: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #35: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #36: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #37: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #38: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #39: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #40: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #41: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #42: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #43: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #44: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #45: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp
```

```
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #47: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #48: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #49: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #50: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #51: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #52: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #53: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #54: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #55: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #56: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #57: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #58: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #59: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #60: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #61: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #62: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #63: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #64: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #65: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #66: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #67: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #68: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #69: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #70: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #71: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #72: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #73: New Button *****
on mouseUp
  put the number of me into it
```

```
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #74: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #75: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #76: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #77: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #78: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #79: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #80: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #81: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #82: New Button *****
on mouseUp
  put the number of me into it
  hide me
```

```
end mouseUp

***** CARD #12, BUTTON #83: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #84: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #85: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #86: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #87: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #88: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #89: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #90: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #91: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp
```

```
***** CARD #12, BUTTON #92: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #93: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #94: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #95: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #96: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #97: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #98: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #99: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #100: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp
```

```
***** CARD #12, BUTTON #101: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #102: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #103: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #104: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #105: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #106: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #107: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #108: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #109: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #111: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #112: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #113: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #114: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #115: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #116: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #117: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #118: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #119: New Button *****
```

```
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #120: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #121: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #122: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #123: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #124: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #125: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #126: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #127: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #128: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #129: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #130: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #131: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #132: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #133: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #134: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #135: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #136: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #137: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #138: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #139: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #140: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #141: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #142: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #143: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #144: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #145: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #146: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
```

```
end mouseUp

***** CARD #12, BUTTON #147: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #148: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #149: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #150: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #151: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #152: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #153: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #154: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #155: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp
```

```
***** CARD #12, BUTTON #156: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #157: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #158: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #159: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #160: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #161: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #162: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #163: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #164: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #165: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #166: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #167: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #168: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #169: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #170: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #171: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #172: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #173: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #174: New Button *****
```

```
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #175: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #176: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #177: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #178: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #179: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #180: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #12, BUTTON #181: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #182: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #183: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #184: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it + 1)
end mouseUp

***** CARD #12, BUTTON #185: New Button *****
on mouseUp
  put the number of me into it
  hide me
  show button (it - 4)
end mouseUp

***** CARD #23, BUTTON #1: anchor *****
on mouseDown
  do_select
end mouseDown

***** CARD #23, BUTTON #2: at_sea *****
on mouseDown
  do_select
end mouseDown

***** CARD #23, BUTTON #3: flag *****
on mouseDown
  do_select
end mouseDown

***** CARD #23, BUTTON #4: overhaul *****
on mouseDown
  do_select
end mouseDown

***** CARD #23, BUTTON #5: inactive *****
on mouseDown
  do_select
end mouseDown

***** CARD #24, BUTTON #1: go_next *****
on mouseUp
  go next
end mouseUp
```

SCRIPTS FOR STACK: compl_report

```
***** BACKGROUND #1: report_graph *****
on openCard
  hide menubar
end openCard

***** BKGND #1, BUTTON #1: print_only_me *****
on mouseUp
  lock screen
  hide bg button print_only_me
  hide bg button go_next
  hide bg button go_prev
  hide bg button go_to_driver
  hide bg button "Performance Report"
  print this card
  show bg button print_only_me
  show bg button lighthouse
  show bg button go_next
  show bg button go_prev
  show bg button go_to_driver
  show bg button "Performance Report"
  unlock screen
end mouseUp

 ***** BKGND #1, BUTTON #2: LightHouse *****
on mouseUp
  go to card 1 of stack PMS_driver
end mouseUp

 ***** BKGND #1, BUTTON #3: go_to_driver *****
on mouseUp
  go to card 2 of stack pms_driver
end mouseUp

 ***** BKGND #1, BUTTON #4: make_me *****
on mouseUp
  repeat with i = 1 to the number of lines of fld collect_info
    put item 1 of line i of fld collect_info into fld i
  end repeat
  repeat with i = 1 to the number of lines of fld collect_info
    if item 2 of line i of fld collect_info = "A" then
      put i * 5 - 4 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "B" then
      put i * 5 - 3 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "C" then
      put i * 5 - 2 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "D" then
      put i * 5 - 1 into WHICH
      show button WHICH
      next repeat
    end if
    if item 2 of line i of fld collect_info = "E" then
      put i * 5 into WHICH
      show button WHICH
    end if
  end repeat
  choose line tool
  put "86," & round(281 - (item 3 of line 1 of fld collect_info)
  * 1.5) into old_loc
  repeat with i = 2 to the number of lines of fld collect_info
    put i * 12 + 74 into h
    put round(281 - (item 3 of line i of fld collect_info) * 1.5) into v
    put h & "," & v into new_loc
    drag from old_loc to new_loc
    put new_loc into old_loc
  end repeat
  choose browse tool
  put "PMS completed :" && average_of (collect_info,3) && "%" into fld total_compl
  put the number of lines of fld collect_info into DAY_NO
  if DAY_NO = 30 then
    show button 156
  else
    if DAY_NO = 29 then
      show button 157
    else
      if DAY_NO = 28 then
        show button 158
      end if
    end if
  end if
end mouseUp

function average_of fld_to,item_on
  put 0 into tot
  put 0 into count
  repeat with i = 1 to the number of lines of fld fld_to
    put tot + item item_on of line i of fld fld_to into tot
    put count + 1 into count
  end repeat
  put (tot / count) into TEMP_RES
  if TEMP_RES - trunc(TEMP_RES) > 0 then
    put (char 1 to (offset(".",TEMP_RES) + 2) of TEMP_RES) into TEMP_RES
  end if
  return (TEMP_RES)
end average_of

 ***** BKGND #1, BUTTON #5: go_next *****
on mouseUp
  go next
end mouseUp

 ***** BKGND #1, BUTTON #6: go_prev *****
on mouseUp
  go prev
end mouseUp


SCRIPTS FOR STACK: General Label Storage
________________________________________________________

****** STACK SCRIPT *****
on closeStack
  if the freesize of this stack > 0.15 * the size of this stack then
    doMenu "Compact Stack"
  end if
end closeStack

***** BACKGROUND #1: bar_code *****
on newCard
  put the date into fld bg_fld_date
end newCard

 ***** BKGND #1, FIELD #6: FSN *****
on returnInField
  put me into fld bar_coded_fld
  click at 135,211
end returnInField

 ***** BKGND #1, FIELD #9: Out_by *****
```

```
on returnInField
  put me into fld bar_coded_fld
  click at 135,211
end returnInField

 ***** BKGND #1, BUTTON #1: print_it *****
on mouseUp
  -- ask the user for the quantity of labels he needs printed
  ask "How many copies do you need ?" with "1"
  -- if he changed his mind...
  if it is "Cancel" or it is empty then
    -- exit event
    exit mouseUp
    -- but if not...
  else
    -- store the copies he needs temporarily
    put it into TIMES_TO_PRINT
    -- suspend screen drawing
    lock screen
    -- hide the buttons
    hide me
    hide bg button bg_but_throw_away
    hide bg button print_and_store
    hide bg button print_and_store_top
    -- do_it for all copies
    repeat with i = 1 to TIMES_TO_PRINT
      print this card
    end repeat
    -- throw card away
    doMenu Oelete Card
    -- go to driver's menu card
    go prev
    -- restore screen drawing
    unlock screen
  end if
  -- exit event
end mouseUp

 ***** BKGND #1, BUTTON #2: bg_but_throw_away *****
on mouseUp
  lock screen
  doMenu Delete Card
  go prev
  unlock screen
end mouseUp

 ***** BKGND #1, BUTTON #3: print_and_store *****
on mouseUp
  -- ask the user to which stack he wants this card stored
  ask "Store this label to:" with "General Label Storage"
  if it is empty or it is "CANCEL" then
    send mouseUp to bg button print_it
  else
    -- set cursor to wait
    set cursor to 4
    -- suspend screen refreshing
    lock screen
    -- check if stack exists
    put it into OEST
    push card
    doMenu "Copy Card"
    go to card 1 of stack OEST
    if the result is not empty then
      unlock screen
```

```
      doMenu "New Stack..."
      lock screen
      doMenu "Paste Card"
      go first
      doMenu "Delete Card"
      pop card
      unlock screen
      send mouseUp to bg button print_it
    else
      doMenu "Paste Card"
      pop card
      unlock screen
      send mouseUp to bg button print_it
    end if
  end if
  -- restore screen drawing
  unlock screen
  -- exit event
end mouseUp

 ***** BKGND #1, BUTTON #4: print_and_store_top *****
on mouseDown
  set the hilite of bg button print_and_store to true
end mouseDown
on mouseUp
  set the hilite of bg button print_and_store to false
  send mouseUp to bg button print_and_store
end mouseUp


SCRIPTS FOR STACK: Generic_fld_print
____________________________________________

 ***** BKGND #1, BUTTON #1: print_me *****
on mouseUp
  global TRANSFERED_OATA
  hide me
  repeat for the number of cards - 1
    doMenu "Delete Card"
  end repeat
  put the number of lines of TRANSFEREO_OATA into TOTAL_LINES
  put 1 into START_LINE
  put 28 into END_LINE
  repeat until START_LINE > TOTAL_LINES
    put line START_LINE to END_LINE of TRANSFEREO_DATA into fld generic_fld
    put END_LINE + 1 into START_LINE
    put START_LINE + 27 into END_LINE
    doMenu "New Card"
  end repeat
  doMenu "Delete Card"
  go first
  print all cards
  show me
end mouseUp


SCRIPTS FOR STACK: Mip_printed
____________________________________________

 ***** BKGND #1, FIELD #7: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #1, FIELD #8: MRC *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #1, FIELD #9: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #1, FIELD #10: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #1, FIELD #11: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #1, FIELD #12: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #1, FIELD #13: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #2, FIELD #4: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #2, FIELD #5: MRC *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #2, FIELD #6: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #2, FIELD #7: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
```

```
end mouseup

 ***** BKGND #2. FIELD #8: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #2, FIELD #9: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #2, FIELD #10: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #3, FIELD #2: test *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #3, FIELD #3: MRC *****
on mouseup
```

```
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #3, FIELD #4: MRCdescript *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #3, FIELD #5: period *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of field "MRC"
  put the selection into cardName
  push card
  go to card cardName
end mouseup

 ***** BKGND #3, FIELD #6: rates *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #3, FIELD #7: man *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp

 ***** BKGND #3, FIELD #8: relman *****
on mouseUp
  put the mousev into it
  put ((the scroll of me) + it) into it
  subtract 96 from it
  put (it div 12) into it
  repeat until line it of field 6 is not empty
    subtract 1 from it
    if it < 1 then exit mouseUp
  end repeat
  put line it of field 6 into it
  push card
  go to card it
end mouseUp


SCRIPTS FOR STACK: movies
_____________________________________________

 ***** BKGND #1, BUTTON #4: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #1, BUTTON #5: New Button *****
on mouseUp
  pop card
end mouseUp

***** CARD #1, BUTTON #1: left *****
on mouseUp
  visual effect dissolve
  go next
end mouseUp

***** CARD #2, BUTTON #1: New Button *****
on mouseUp
  visual effect dissolve
  go prev
end mouseUp


 SCRIPTS FOR STACK: PMS Toolbox _______________________________________________

***** BACKGROUND #1: MIP-1st page *****
function fieldsort thefield,sort_on_item
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield - 1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines in thefield
      if item sort_on_item of line current of thefield < item sort_on_item of line least
linenum of thefield then
        put current into leastlinenum
      end if
    end repeat
    if leastlinenum > primary then
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsort
```

```
 ***** BKGND #1, BUTTON #1: LightHouse *****
on mouseUp
  go to stack PMS_driver
end mouseUp

 ***** BKGND #1, BUTTON #3: return *****
on mouseUp
  pop card
end mouseUp

***** BACKGROUND #2: MIP-1st page *****

 ***** BKGND #2, FIELD #3: MIP LIST *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of me
  put the selection into cardName
  push card
  visual effect scroll left
  push card
  go to stack cardName    --new code
end mouseup

 ***** BKGND #2, BUTTON #1: LightHouse *****
on mouseUp
  go first
end mouseUp

***** CARD #1, BUTTON #1: append *****
-- This function takes a list of stacks and adds them to the end of
-- the PMS database.
on mouseUp
  -- prepare holder for MIP to append
  global DEST
  -- verify with the user that he realy wants to procceed adding a MIP set
  answer "Procede Adding a New Set ?" with "Cancel" or "Ok"
  -- if he does...
  if it is "Ok" then
    -- do it for all MIPs in the list
    repeat with new_mips = the number of lines of card fld mip_in down to 1
      -- preserve this card (toolbox card)
      push card
      -- inform the user
      show bg button "Working..."
      -- pick up the last satck in the list to be appended and store
      -- it in a global variable
      put line new_mips of card fld mip_in into DEST
      -- if nothing is there...
      if DEST is empty or DEST is return then
        -- pick up next
        next repeat
      end if
      -- but if something is there...
      -- report it into the field proccessed
      put DEST & return before card fld mip_out
      -- sort the field for easy viewing
      put fieldsort(card field mip_out,1) into card field mip_out
      -- remove it from the list of the ones to be proccessed
      delete line new_mips of card fld mip_in
      -- preserve toolbox card
      push card
      -- suspend screen drawing for faster execution
      lock screen
      -- go to this stack
      go to stack DEST
      -- if it is not found...
      if the result is not empty then
        -- go back to toolbox
        pop card
        -- pick up next MIP/MRC set to be proccessed
        next repeat
        -- but if it is there...
      else
        -- go back
        pop card
      end if
      -- go to the PMS database
      go to stack PMS
      -- send event to the append button that is there top avoid back
      -- and forth stack swapping
      send mouseUp to bg button append
      -- come back to toolbox
      pop card
      -- restore screen to inform user
      unlock screen
    end repeat
    -- preserve card
    push card
    -- suspend screen drawing
    lock screen
    -- go to the PMS database
    go to stack PMS
    -- inform uaer
    put "Compacting Stack ..............." into msg
    -- compact the PMS database to make it perform faster
    doMenu "Compact Stack"
    -- return to the toolbox
    pop card
    -- flag out the updated field of the list card
    put "NO" into card fld updated of card mip_mrc_list
    -- restore acreen
    hide bg button "Working..."
    unlock screen
    -- click outside the fields
    click at 31,178
    -- hide the message box
    hide msg
  else
    -- if the user does not want to procceed...Exit event
    exit mouseUp
  end if
  -- get user's attention, to let him know the opperation has been completed
  play "Mozart"
  -- exit event
end mouseUp

***** CARD #1, BUTTON #2: print_this_fld_add_mip *****
on mouseDown
  global TRANSFERED_DATA
  put "MIPs in" into popup
  put return & "MIPa out" after popup
  get HPopupMenu(popup,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      put "card fld mip_in" into print_what
      put "MIPs TO ADD" & return into temp_label
    end if
    if theline = 2 then
      put "card fld mip_out" into print_what
      put "MIPs ADDED" & return into temp_label
    end if
    put "PRINTED ON:" &"     " & the date &"     " & the time & return
    & return into TRANSFERED_DATA
    put temp_label & return & return after TRANSFERED_DATA
    put the value of print_what after TRANSFERED_DATA
    set cursor to 4
    push card
    lock screen
    go to stack generic_fld_print
    send mouseUp to bg button print_me
    pop card
    unlock screen
    put empty into TRANSFERED_DATA
  end if
end mouseDown

***** CARD #2, BUTTON #1: New Button *****
on mouseUp
  -- inform user
  show bg button "Working..."
  -- set cursor to wait
  set cursor to 4
  -- set new location for the msg box
  set the loc of msg to 15,247
  -- empty the field to report newly missing MIPs
  put empty into card fld mip_not_found
  -- repeat for all the MIPs for this ship
  repeat with i = 1 to the number of lines of card fld mip_in
    -- mlock the screen to let things roll faster
    lock screen
    -- check if the job has been proccessed before and if not...
    if char 1 of line i of card fld mip_in is not "*" then
      -- calculate how many work centera are involved to thia MIP
      put the number of items of line i of card fld mip_in into WC_PLUS_1
      -- extract the MIP number
      put item 1 of line i of card fld mip_in into MIP_EXAMINED
      -- save temporarily the whole line
      put line i of card fld mip_in into MIP_LINE
      -- in form the user which MIP is getting proccessed
      put "Proccessing MIP #" && MIP_EXAMINED into msg
      -- preserve the toolbox card
      push card
      -- go to the PMS database
      go to stack PMS
      -- look if this MIP is in the system
      find MIP_EXAMINED in fld MIP
      -- if it is, we are at the first page of this MIP card
      if the result is empty then
        -- move to the second page of the MIP card
        go next
        -- ...and from here repeat
        repeat
          -- move to the next page of the MRC
          go next
          -- check to recognize that it is the first page of the MRC card
          if the short name of this bg is "MRC - 1st page." then
            -- check if this MRC belongs to this MIP
            if MIP_EXAMINED is not in line 1 of fld data then
              -- ...and if not, exit the repeat loop
            exit repeat
            -- ...but if it belongs
          else
            -- clear the work center loop from previous attempts
            put empty into fld other_wc_involved
            -- repeat for all the work centers in the list
            repeat with iii = 2 to WC_PLUS_1
              -- put each work center in a new line of this field
              put item iii of MIP_LINE & return after fld other_wc_involved
            end repeat
          end if
        end if
      end repeat
      -- go back to the toolbox to report actions
      pop card
      -- mark this MIP "proccessed
      put "*" before line 1 of card fld mip_in
      -- but if it was not found...
    else
      -- get back to the toolbox
      pop card
      -- report this MIP card as not found
      put MIP_EXAMINED & return after card fld mip_not_found
      -- restore screen to show changes
      unlock screen
    end if
  end if
end repeat
-- relock the screen to let things happen at the background
lock screen
-- preserve this card
push card
-- prepare new variable
put empty into CHECK_IF_NO_WC
-- go to the PMS database to check if all cards were proccessed
go to stack PMS
-- do it for all MRC cards
repeat with i = 1 to the number of cards of bg "MRC - 1st page."
  -- examine the work shops involved field
  get fld other_wc_involved of card i of bg "MRC - 1st page."
  -- if no work centers are there
  if it is empty then
    -- find to what MIP this MRC belongs to
    get line 1 of fld data of card i of bg "MRC - 1st page."
    -- remove the first 4 characters to get only its number
    delete char 1 to 4 of it
    -- if it is not allready reported as missing
    if it is not in CHECK_IF_NO_WC then
      -- put it into the variable to report it
      put it  & return after CHECK_IF_NO_WC
    end if
  end if
end repeat
-- compact free space to return the space to the system
doMenu "Compact stack"
-- return to the toolbox
pop card
-- if no bad MRCs were found
if CHECK_IF_NO_WC is empty then
  -- report it as success
  put "ALL MRCs OK !!!" into card fld not_proccessed
  -- but if there were some...
else
  -- put the results of the search into the appropriate fld
    put CHECK_IF_NO_WC into card fld not_proccessed
  end if
  -- play tune to get attention !!!
  play "MOZART"
  -- restore the screen
  hide bg button "Working..."
  -- restore the position of the message box
  set the loc of msg to 22,300
  -- hide the message box
  hide msg
  -- restore screens condition
  unlock screen
  -- exit event
end mouseUp

***** CARD #2, BUTTON #2: print_this_fld_add_mip *****
on mouseDown
  global TRANSFERED_DATA
  put "MIP-WC list" into popup
  put return & "MIP not in PMS db" after popup
  put return & "Not processed MIP" after popup
  get HPopupMenu(popup,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      put "card fld mip_in" into print_what
      put "WC RESPONSIBILITY LIST" & return into temp_label
    end if
    if theline = 2 then
      put "card fld mip_not_found" into print_what
      put "MIP NOT IN PMS DB" & return into temp_label
    end if
    if theline = 3 then
      put "card fld not_proccessed" into print_what
      put "MIP WITH NO WORK CENTER RESPONSIBLE FOR" & return into temp_label
    end if
    put "PRINTED ON:" &"     " & the date &"     " & the time & return
    & return into TRANSFERED_DATA
    put temp_label & return & return after TRANSFERED_DATA
    put the value of print_what after TRANSFERED_DATA
    set cursor to 4
    push card
    lock screen
    go to stack generic_fld_print
    send mouseUp to bg button print_me
    pop card
    unlock screen
    put empty into TRANSFERED_DATA
  end if
end mouseDown

***** CARD #3, BUTTON #1: del_mips *****
-- This function takes a list of MIP cards and goes in the PMS database
-- and deletes them along with their coresponding MRCs.  It is used for
-- removing a MIP to be able to add a newer version of it.
on mouseUp
  -- inform the user
  show bg button "Working..."
  -- set cursor to wait
  set cursor to 4
  -- suspend scren drawing
  lock screen
  -- clear the field to report deleted MIPs
  put empty into card fld mip_del
  -- do it for all MIPs to be deleted
  repeat with i = the number of lines of card fld mip_in down to 1
    -- get the MIP to be deleted
    put line 1 of card fld mip_in into MIP_EXAMINED
    -- preserve this card
    push card
    -- find the MIP card to be deleted
    go to card MIP_EXAMINED of stack PMS
    -- if this card is not there
    if the result is not empty then
      -- return to the preserved card to pick next MIP
      pop card
      -- ...but if it is
    else
      -- delete both pages of it
      doMenu "Delete Card"
      doMenu "Delete Card"
      -- delete also all cards after this until you meet next MIP
      repeat until the short name of this bg is "MIP-1st page"
        doMenu "Delete Card"
      end repeat
      -- return to preserved card
      pop card
      --report this Mip as deleted
      put MIP_EXAMINED & return after card fld mip_del
      -- remove it from the ones to delete
      delete line 1 of card fld mip_in
    end if
  end repeat
  -- preserve toolbox card
  push card
  -- go to the PMS database and compact it to give back to the system the freed space
  go to stack PMS
  doMenu "Compact stack"
  -- return to toolbox
  pop card
  -- call the user's attention
  play "MOZART"
  -- restore screen
  hide bg button "Working..."
  unlock screen
  -- exit event
end mouseUp

***** CARD #3, BUTTON #2: print_this_fld_add_mip *****
on mouseDown
  global TRANSFERED_DATA
  put "MIP to remove from PMS db" into popup
  put return & "MIP removed from PMS db" after popup
  get HPopupMenu(popup,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      put "card fld mip_in" into print_what
      put "MIP TO REMOVE FROM PMS DB" & return into temp_label
    end if
    if theline = 2 then
      put "card fld mip_del" into print_what
      put "MIP REMOVED FROM PMS DB" & return into temp_label
    end if
    put "PRINTED ON:" &"    " & the date &"    " & the time & return
    & return into TRANSFERED_DATA
    put temp_label & return & return after TRANSFERED_DATA
    put the value of print_what after TRANSFERED_DATA
```

```
    set cursor to 4
    push card
    lock screen
    go to stack generic_fld_print
    send mouseUp to bg button print_me
    pop card
    unlock screen
    put empty into TRANSFERED_DATA
  end if
end mouseDown

***** CARD #4: file to stack *****
on closecard
  global check_var
  if check_var = true then
    exit closecard
  end if
  if  field "MIP LIST" = empty then
    exit closecard
  end if
  set lockscreen to true
  put "1" into count
  put "1" into count1
  repeat for the number of lines of  field "MIP LIST"
    put line count of  field "MIP LIST" &" " & line count of
    card field "MIP NAME" into line count of
    holder
    put count + 1 into count
  end repeat
  put empty into field "MIP LIST"
  put empty into card field "MIP NAME"
  put fieldsort (holder) into holder
  repeat for the number of lines of holder
    put the number of words of line count1 of holder
    into test
    put word 1 of line count1 of holder
    into line count1 of field "MIP LIST"
    put word 2 to test of line count1 of holder
    into line count1 of card field "MIP NAME"
    put count1 + 1 into count1
  end repeat
  put return after line count1-1 of  field "MIP LIST"
  put return after line count1-1 of card field "MIP NAME"
  set lockscreen to false
  put true into check_var
end closecard

Function fieldsort thefield
  put return & return into crcr
  get thefield
  repeat
    put offset(crcr,it) into p
    if p = 0 then exit repeat
    delete char p of it
  end repeat
  put it into thefield
  repeat with primary = 1 to the number of lines in thefield -1
    put primary into leastlinenum
    repeat with current = primary + 1 to the number of lines
      in thefield
      if line current of thefield < line leastlinenum of
      thefield then
        put current into leastlinenum
    end repeat
    if leastlinenum > primary then
      put line primary of thefield into temp
      put line leastlinenum of thefield into line primary
      of thefield
      put temp into line leastlinenum of thefield
    end if
  end repeat
  return thefield
end fieldsort

on idle
  set the scroll of field "MIP LIST" to scroll of card field "MIP NAME"
  -- put the number of lines in card field "MIP NAME" into card
  --field "number of mips"
end idle

***** CARD #4, FIELD #1: MIP NAME *****
on mouseup
  get item 2 of the clickloc - top of me + scroll of me
  put 1 + it div the textheight of me into LineNum
  select Line LineNum of card field "MIP LIST"
  put the selection into cardName
  push card
  visual effect scroll left
  go to stack cardName
end mouseup

***** CARD #4, FIELD #4: back *****
on mouseup
  hide me
end mouseup


***** CARD #4, BUTTON #1: Add MIP/MRCs *****
on mouseUp
  -- This procedure is used to produce both MIP card and MRC card using
  -- files received from the navy's PMS data base
  -- All that is regulred prior running this procedure on a file received
  -- from the PMS database is to ensure that:
  -- 1. The file consists of only one MIP and its associated MRCs
  -- Change all " to blanks
  -- 3. Change the last ===> in the file to STOP

    global check_var,procedure1,hdtemp
  if hdtemp = empty then
    axit mouseup
  end if
  lock screen
  push card
  go to card "mip page 1 template" of stack "templates"
  domenu "NEW STACK..."
 set the name of this stack to "Temp"
  pop card
  unlock acreen
  put tha number of lines of fiald list into listemp
  repeat with i = 1 to the number of lines of field "list"
    put listemp - 1 into card field "number of MIPS"
    if hdtemp = empty then
      exit mouseup
    end if
    put hdtemp into infile
    put line 1 of field "list" into it
    set cursor to watch
```

```
put it after INFILE
go to card "file to stack" of this stack
lock screen
show card field prompt
unlock screen with visual wipe right
open file infile
read from file infile until return
repeat until (char 3 to 5 of it = "MIP")
  read from file infile until return
end repeat
put char 11 to 21 of it into MipNom
find MipNom in field "Mip List" of card "file to stack"
-- "Looking to see if information is already in the database"
if the result = empty then
  put listemp - 1 into listemp
  close file infile
  next repeat
end if
put char 61 to 74 of it into cdate
put empty into config1
repeat until (char 4 to 14 of it = "Ship System")
  read from file infile until return
end repeat
repeat until (char 3 of it <> " ")
  read from file infile until return
end repeat
put char 3 to 50 of it & return into MipNam
put char 3 to 50 of it & return into MipName
put char 65 to 90 of it & return into Pub
repeat until ((char 3 of it = " ") or (char 65 of it = " "))
  read from file infile until return
  put char 3 to 50 of it after MipName
  put char 3 to 50 of it after MipNam
  put char 65 to 90 of it & return after Pub
end repeat
repeat until ((char 4 to 16 of it = "CONFIGURATION") or
  (char 3 to 5 of it = "T O"))
  read from file infile until return
end repeat
if (char 4 to 16 of it = "CONFIGURATION") then
  read from file infile until return
  repeat until (char 3 to 5 of it = "T O")
    if (char 3 to 5 of it = "T O") then
      exit repeat
    end if
    read from file infile until return
    if (char 3 to 5 of it = "T O") then
      exit repeat
    end if
    put char 3 to 110 of it & return after config1
  end repeat
end if
repeat until (char 3 to 5 of it = "T O")
  read from file infile until return
end repeat
read from file infile until return
read from file infile until return
put falae into inactive_test
put empty into in_MRC
put empty into act_MRC
put empty into in_MRC_data
put empty into MRC_data
put " " & return after test1
put " " & return after MRC1
```

```
  put " " & return after descript
  put " " & return after pl
  put " " & return after rl
  put " " & return after ml
  put " " & return after rell
  put 0 into tapetest
  repeat until (char 3 to 9 of it = "END MIP")
    if char 3 to 9 of it = "END MIP" then
      exit repeat
    end if
    if char 11 of it <> " " then
      add 1 to tapetest
    end if
    if char 10 of it = "-" then
      put "." into char 10 of it
    end if
    if char 33 to 62 of it = "INACTIVE EQUIPMENT MAINTENANCE" then
      put true into inactive_test
    end if
    if inactive_test then
      if char 11 of it <> " " then
        put char 10 to 19 of it & return after in_MRC
      end if
    else
      if char 11 of it <> " "then
        put char 10 to 19 of it & return after act_MRC
      end if
    end if
    if ((char 11 of it <> " ") or (char 76 of it <> " ") or
    (char 94 of it <> " ")) then
      put char 10 to 19  of it & " " into MRC2
      put char 76 to 78  of it &  " " into p2
      put char 94 to 96  of it into m2temp
      if (m2temp = "Var" or m2temp = "ies") then
        put "     " into m2
      else
        put char 94 to 96  of it & " " into m2
      end if
      put char 99 to 102  of it &  " " into rel2
      if inactive_test then
        put MRC2 & p2 & m2 & rel2 & return after in_MRC_data
      else
        put MRC2 & p2 & m2 & rel2 & return after MRC_data
      end if
    end if
    put char 3  of it & return after test1
    if char 10 of it = "." then
      put char 10 to 19 of it & return after MRC1
    else
      put char 11 to 19 of it & return after MRC1
    end if
    put char 23 to 72 of it & return after descript
    put char 76 to 78  of it & return after pl
    put char 84 to 89  of it & return after rl
    put char 94 to 96  of it into mitemp
    if (mitemp = "Var" or mitemp = "ies") then
      put "" & return after ml
    else
      put char 94 to 96  of it & return after ml
    end if
    put char 99 to 102  of it & return after rell
    read from file infile until return
  end repeat
  put MipNom & return after  field "MIP LIST"
```

```
put MipName & return after card field "MIP NAME"
put false into check_var
read from file infile until return
lock screen
go to stack "templates"
go to card "mip page 1 template"
doMenu "COPY CARD"
push card
go to stack temp
doMenu "LAST"
doMenu "PASTE CARD"
set the name of this card to MipNom
put MipName into field "card_NAME"
put MipNom into field "MIP"
put cdate into field "card_date"
put MipName into field "ship_system"
put pub into field "pubs"
put config1 into field "config"
pop card
go to card "mip page 2 template" of this stack
doMenu "COPY CARD"
push card
go to stack temp
doMenu "LAST"
doMenu "PASTE CARD"
set the name of this card to MipNom & "-1"
put in_MRC into field MRC_of_this_MIP_inactive
put act_MRC into field MRC_of_this_MIP_active
put MRC_data into field data_MRC
put in_MRC_data into field in_data_MRC
put MipName into field "card_NAME"
put MipNom into field "MIP"
put test1 into field "test"
put MRC1 into field "MRC"
put descript into field "MRCdescript"
put pl into field "period"
put rl into field "rates"
put ml into field "man"
put rell into field "relman"
pop card
put tapetest into tt
repeat forever
  push card
  go to card "file to stack" of stack "PMS Toolbox"
  show card field "number of MIPS"
  hide card field working
  show card field res
  show card field res1
  show card field MRCn
  show card field MRCm
  show card field back
  show card field MIPm
  show card field "MIPnumber"
  show card field "MIP"
  show card field "numberm"
  show card field workingon
  put empty into card field res
  put empty into card field res1
  put empty into card field "MIPnumber"
  put the number of lines of field list into card field "numberm"
  put tt into card field res1
  if tapetest <> 0 then
    put tapetest - 1 into tapetest
put tapetest into card field res
put MipNom into card field MIPnumber
unlock screen with iris open
lock screen
pop card
repeat until  ((char 3 to 5 of it = "MRC") or
  (char 1 to 4 of it = "STOP"))
  read from file infile until return
end repeat
if ((char 1 to 4 of it  = "STOP")) then
  exit repeat
end if
put char 7 to 13 of it into MRC_number
go to stack templates
go to card "mrc card 1 template"
doMenu "COPY CARD"
go to stack temp
doMenu "LAST"
doMenu "PASTE CARD"
put 1 into field page_no
put "MIP" && MipNom & return into field data
put "LOG" & return after field data
put char 1 to 2 of MRC_number && char 3 to 6 of MRC_number &&
char 7 of MRC_number into name1
put name1 into field MRCno
set the name of this card to name1
put char 61 to 73 of it into date_number
repeat until ((char 4 to 14 of it = "SHIP SYSTEM") or
  (char 4 to 9 of it = "SYSTEM") or (char 61 to 68 of
  it = "MRC CODE"))
  read from file infile until return
end repeat
read from file infile until return
read from file infile until return
put char 62 to 65 of it into code
put char 69 to 71 of it into cycle
repeat until ((char 4 to 9 of it = "SYSTEM") or (char 33 to 41 of
  it = "EQUIPMENT") or ( char 61 to 65 of it = "RATES"))
  if char 3 of it <> " " then
    put char 3 to 29 of it & return after ship
  end if
  if char 33 of it <> " " then
    put char 33 to 57 of it & return after subsystem
  end if
  read from file infile until return
end repeat
read from file infile until return
read from file infile until return
put char 3 to 29 of it & return after system
put char 33 to 57 of it & return after equip
put char 72 to 74 of it into ratestemp
if ratestemp = "Var" or ratestemp = "les" then
  put char 61 to 66 of it & "   " & return into ratea
else
  put char 61 to 66 of it & char 70 to 74 of it & return into rates
end if
read from file infile until return
repeat until (char 64 of it = "T") or (char 4 to 14 of it = "MAINTENANCE")
  if (char 4 to 14 of it = "MAINTENANCE") then
    exit repeat
  end if
  if char 3 of it <> " " then
    put char 3 to 29 of it & return after system
  end if
```

```
  if char 33 of it <> " " then
    put char 33 to 60 of it & return after equip
  end if
  if char 61 of it <> " " then
    put char 72 to 74 of it into ratestemp
    if ratestemp = "Var" or ratestemp = "les" then
      put char 61 to 66 of it & "   " & return into rates
    else
      put char 61 to 66 of it & char 70 to 74 of it & return into rates
    end if
  end if
  read from file infile until return
end repeat
if (char 64 of it = "T") then
  read from file infile until return
  put char 69 to 71 of it into MANhourstemp
  if (MANhourstemp = "Var" or MANhourstemp = "les") then
    put "" into MANhours
  else
    put char 69 to 71 of it into MANhours
  end if
  read from file infile until return
  read from file infile until return
  put char 69 to 71 of it into elapsed1temp
  if (elapsed1temp = "Var" or elapsed1temp = "les") then
    put "" into elapsed1
  else
    put char 69 to 71 of it into elapsed1
  end if
end if
put empty into process
repeat until (char 4 to 14 of it = "MAINTENANCE")
  read from file infile until return
end repeat
repeat while (char 4 to 9 of it <> "SAFETY")
  read from file infile until return
  if char 4 to 9 of it = "SAFETY" then
    exit repeat
  else
    if char 2 to 10 of it <> "         " then
      put char 2 to 74 of it & return after process
    end if
  end if
end repeat
put MipName into field what_equipment
put date_number into field date_edited
put ship into field ship_system_field
put subsystem into field subsystem_field
put code into MIP
put system into field system_field
put equip into field equipment_field
put rates into field rates_mh_field
put process into field mr_description_field
put manhours into field MH
put elapsed1 into field elapsed
put rates into field rates_mh_field
put code into field mrcode
put cycle into field periodicity
repeat until (char 3 to 5 of it = "END")
  if char 3 to 5 of it = "END" then
    exit repeat
  else
    if char 4 to 21 of it = "SAFETY PRECAUTIONS" then
      put "==========================================================================
----------------" & return after procedure1
            put char 2 to 78 of it & return after procedure1
            put "---------------------------------------------------------------------------
----------------" & return after procedure1
          else if char 4 to 15 of it = "TOOLS, PARTS" then
            put "---------------------------------------------------------------------------
----------------" & return after procedure1
            put char 2 to 78 of it & return after procedure1
            put "---------------------------------------------------------------------------
----------------" & return after procedure1
          else if (char 4 to 12 of it = "PROCEDURE") then
            put "---------------------------------------------------------------------------
----------------" & return after procedure1
            put char 2 to 78 of it & return after procedure1
            put "---------------------------------------------------------------------------
----------------" & return after procedure1
          else if char 4 to 9 of it = "TOOLS " then
            put char 4 to 9 of it into char 5 to 10 of it
            put "¥  " into char 2 to 4 of it
            put char 2 to 78 of it & return after procedure1
          else if char 4 to 8 of it = "TEST " then
            put char 4 to 8 of it into char 5 to 9 of it
            put "¥  " into char 2 to 4 of it
            put char 2 to 78 of it & return after procedure1
          else if (char 4 to 12 of it = "MATERIALS") then
            put char 4 to 12 of it into char 5 to 14 of it
            put "¥  " into char 2 to 4 of it
            put char 2 to 78 of it & return after procedure1
          else if (char 4 to 16 of it = "MISCELLANEOUS") then
            put char 4 to 16 of it into char 5 to 17 of it
            put "¥  " into char 2 to 4 of it
            put char 2 to 78 of it & return after procedure1
          else if (char 3 to 9 of it = "WARNING") then
            put "¥" into char 1 of it
            put char 1 to 78 of it & return after procedure1
          else if (char 3 to 13 of it = "Preliminary") then
            put char 3 to 13 of it into char 5 to 15 of it
            put "¥  " into char 2 to 4 of it
            put char 2 to 78 of it & return after procedure1
          else
            put char 2 to 78 of it & return after procedure1
          end if
        end if
        read from file infile until return
      end repeat
      put "            " & return after procedure1
      put "---------------------------------------------------------------------------
----------" & return after procedure1
      put the number of lines of procedure1 into proc_test
      put proc_test div 300 into full_pages
      put proc_test mod 300 into remaining_pages
      if remaining_pages > 0 then
        put full_pages + 1 into full_pages
      end if
      put 1 into first_line
      put 300 into last_line
      put (full_pages - 1) into count
      put count & ",0,0" into line 4 of field data
      put "ARGOS" into line 5 of field data
      put "MOVIE" into line 6 of field data
      put 2 into page
      put 1 into page_count
      repeat for full_pages
          go to card "mrc card 2 template"
          doMenu "COPY CARD"
          go to stack temp
          doMenu "LAST"
          doMenu "PASTE CARD"
          put MipName into field what_equipment
          set the name of this card to name1 &"-"& page_count
          put page into field page_no
          put name1 into field MRCno
          if last_line > proc_test then
            put line first_line to proc_test of procedure1 ¬
            into field procedure
          else
            put line first_line to last_line of procedure1 ¬
            into field procedure
          end if
          put last_line + 1 into first_line
          put last_line + 300 into last_line
          put (page + 1 )into page
          put page_count + 1 into page_count
        end repeat
        put empty into procedure1
        put empty into date_number
        put empty into  ship
        put empty into  subsystem
        put empty into  code
        put empty into system
        put empty into  equip
        put empty into  rates
        put empty into  process
        put empty into  manhours
        put empty into  elapsed1
        put empty into  rates
        put empty into  code
        put empty into  cycle
        put empty into  MRC_number
        put empty into  name1
        put empty into page_count
        put empty into MIP
      end repeat
      close file infile
      go to card "file to stack" of stack "PMS Toolbox"
      put listemp - 1 into listemp
      if listemp <> 0 then
        show card field working
      end if
      hide card field res
      hide card field res1
      hide card field MRCn
      hide card field MRCm
      hide card field "MIPnumber"
      hide card field workingon

      unlock screen with iris open
      put empty into test1
      put empty into MRC1
      put empty into descript
      put empty into p1
      put empty into r1
      put empty into m1
      put empty into rel1
    end repeat
    show card field "the end"
```

```
  push card
  put the long time into datevar
  convert datevar to dateitems
  put " " & item 4 of datevar & "/" & item 5 of datevar into datevar
  go to stack "temp"
  set the name of this stack to "New PMS " & the short date & datevar
  domenu "First"
  domenu "Delete Card"
  domenu "Compact Stack"
  pop card
  unlock screen
  lock screen
  hide card field "the end"
  hide card field working
  hide card field "MIP"
  hide card field back
  hide card field MIPm
  hide card field "numberm"
  hide card field prompt
  hide card field "number of MIPS"
  unlock screen with iris open
  beep 5
end mouseUp

***** CARD #4, BUTTON #2: clear text files *****
on mouseUp
  global hdtemp
  put empty into hdtemp
  put empty into card field hardpath
  put empty into field list
end mouseUp

***** CARD #4, BUTTON #3: get path name *****
on mouseUp
  global hdtemp
  put 0 into count
  put filename("TEXT") into hdtemp
  put the number of chars of hdtemp into temp
  repeat with j = temp down to 1
    if char j of hdtemp = ":" then
      put j into count
      exit repeat
    end if
  end repeat
  put char 1 to count of hdtemp into hdtemp
  put hdtemp into card field hardpath
end mouseUp

***** CARD #4, BUTTON #4: LOCK/UNLOCK *****
on mouseUp
  set the locktext of  field "mip list" to
  not (the locktext of  field "mip list")
  set the locktext of  card field "mip name" to
  not (the locktext of  card field "mip name")
end mouseUp

***** CARD #4, BUTTON #5: get text files *****
on mouseUp
  global tftemp,hdtemp
  put 0 into count
  put filename("TEXT") into tftemp
  if tftemp is empty then
    exit mouseup
  end if
  put the number of chars of tftemp into temp
  repeat with j = temp down to 1
    if char j of tftemp = ":" then
      put j + 1 into count
      exit repeat
    end if
  end repeat
  put char 1 to count - 1 of tftemp into hdtemp
  put hdtemp into card field hardpath
  put char count to temp of tftemp into tftemp
  find tftemp in field list
  if the result = empty then
    beep
    exit mouseup
  else
    put tftemp & return after field list
  end if
end mouseUp

***** CARD #4, BUTTON #6: print_this_fld_add_mip *****
on mouseUp
  global TRANSFERED_DATA
  put empty into TEMP_TRANSFERED_DATA
  repeat with i = 1 to the number of lines of fld "MIP LIST"
    put line i of fld "MIP LIST" && "." && line i of card
    fld "MIP NAME" & return after TEMP_TRANSFERED_DATA
  end repeat
  put "PRINTED ON:" &"     " & the date &"     " & the time & return
  & return into TRANSFERED_DATA
  put "MIP #        NOMENCLATURE" & return & return after TRANSFERED_DATA
  put TEMP_TRANSFERED_DATA after TRANSFERED_DATA
  --set cursor to 4
  push card
  lock screen
  go to stack generic_fld_print
  send mouseUp to bg button print_me
  pop card
  unlock screen
  put empty into TRANSFERED_DATA
end mouseUp

***** CARD #5: empty_PMS_db *****
on returnkey
  if the shiftkey is down then
    pass returnkey
  else
    show bg button "Working..."
    lock screen
    set the loc of msg to 22,-1500
    put msg into PRESERVE_PWD
    put empty into msg
    hide msg
    set the loc of msg to 22,300
    push card
    go to stack "Password Stack"
    find PRESERVE_PWD in fld dwp
    if the result is not empty then
      deny_access
      exit returnkey
    else
      if fld bg_fld_position is not "Super User" then
        deny_access
        exit returnkey
      else
```

```
        pop card
        unlock screen
        answer "This opperation will delete the PMS db" with "Delete"
        or "Cancel"
        if it is "Delete" then
          answer "Sure...This is the last chance..." with "Proceed" or
          "Cancel"
          if it is "Proceed" then
            set cursor to 4
            lock screen
            go to card 8 of stack PMS
            repeat with i = 8 to the number of cards
              doMenu "Delete Card"
            end repeat
            doMenu "Compact Stack"
            pop card
            pop card
            unlock screen
          else
            pop card
            hide bg button "Working..."
            unlock screen
            exit returnkey
          end if
        else
          pop card
          hide bg button "Working..."
          unlock screen
          exit returnkey
        end if
      end if
    end if
  end if
end returnkey

on deny_access
  pop card
  hide bg button "Working..."
  pop card
  show button "Access denied !!!"
  beep
  unlock screen
  wait 4 seconds
  hide button "Access denied !!!"
end deny_access

***** CARD #5, BUTTON #1: del_PMS_db *****
on mouseUp
  -- prompt the user to enter his password
  -- if the correct password is supplied, the event is sent to the
  -- card levelto perform the opperation.
  play "PASSWORD PLEASE"
  -- exit event
end mouseUp

***** CARD #6, BUTTON #1: append *****
-- This function makes a list of all the MIP and MRC cards in the
-- PMS database.
on mouseUp
  -- preserve this card (toolbox card)
  push card
  -- set the cursor to wait
  set cursor to 4
  -- inform the user
```

```
  show bg button "Working..."
  -- suspend screen drawing to speed up execution
  lock screen
  -- prepare temporary variables
  put empty into MIPS
  put empty into MRCS
  -- go to the PMS database
  go to stack PMS
  -- do it for all the MIP cards
  repeat with i = 1 to the number of cards of bg "MIP-1st page"
    -- get their name
    get fld MIP of card i of bg "MIP-1st page"
    -- store it in the temporary container
    put it & return after MIPS
  end repeat
  -- do it now for all the MRC cards of the database
  repeat with i = 1 to the number of cards of bg "MRC - 1st page."
    -- get their name
    get fld MRCno of card i of bg "MRC - 1st page."
    -- store it in the temporary container
    put it & return after MRCS
  end repeat
  -- go back to the toolbox
  pop card
  -- remove the template card names
  delete line 1 of MIPS
  delete line 1 of MRCS
  -- put the variables into the appropriate fields
  put MIPS into card fld mip_numbers
  put MRCS into card fld mrc_numbers
  -- count the MIP cards
  put the number of lines of card fld mip_numbers into
  card fld total_mip
  -- count the MRC cards
  put the number of lines of card fld mrc_numbers into
  card fld total_mrc
  -- set the flag to updated
  put "YES" into card fld updated
  -- restore the screen
  hide bg button "Working..."
  unlock screen
  -- get users attention
  play "MOZART"
  -- exit event
end mouseUp

***** CARD #6, BUTTON #2: print_this_fld_add_mip *****
on mouseDown
  global TRANSFERED_DATA
  put "MIP in PMS db" into popup
  put return & "MRC in PMS db" after popup
  get HPopupMenu(popup,0,the mouseV,the mouseH)
  if it is not zero then
    put item 1 of it into theline
    if theline = 1 then
      put "card fld mip_numbers" into print_what
      put "MIP CARDS IN PMS DB" & return into temp_label
    end if
    if theline = 2 then
      put "card fld mrc_numbers" into print_what
      put "MRC CARDS IN PMS DB" & return into temp_label
    end if
    put "PRINTED ON:" &"    " & the date &"    " & the time & return
    & return into TRANSFERED_DATA
    put temp_label & return & return after TRANSFERED DATA
    put the value of print_what after TRANSFERED DATA
    set cursor to 4
    push card
    lock screen
    go to stack generic_fld_print
    send mouseUp to bg button print_me
    pop card
    unlock screen
    put empty into TRANSFERED_DATA
  end if
end mouseDown


SCRIPTS FOR STACK: Ship_data
```

```
****** STACK SCRIPT *****
on closeStack
  if the freesize of this stack > 0.15 * the size of this stack then
    doMenu "Compact Stack"
  end if
end closeStack

function centered item_to_center,total_length
  -- returns into item_to_center, a string of length <total_length> with
  -- original item_to_center, centered in it.
  put trunc ((total_length - length(item_to_center)) / 2) into fill_beg
  repeat with i = 1 to fill_beg
    put " " before item_to_center
  end repeat
  repeat with i = 1 to (total_length - length(item_to_center))
    put " " after item_to_center
  end repeat
  return item_to_center
end centered

function left_justified item_to_left,total_length
  -- returns into item_to_left, a string of length <total_length> with
  -- original item_to_left, left justified in it.

  repeat with i = 1 to (total_length - length(item_to_left))
    put " " after item_to_left
  end repeat
  return item_to_left
end left_justified

***** BACKGROUND #1 *****

 ***** BKGND #1, BUTTON #3: LightHouse *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #2, BUTTON #1: New Button *****
on mouseUp
  ask "Search for what ?"
  if it is empty or it is "CANCEL" then
    exit mouseUp
  else
    put it into FIND_WHAT
    push card
    find FIND_WHAT
    if the result is not empty then
      beep
      answer "String not found"
      exit mouseUp
    end if
  end if
end mouseUp

 ***** BKGND #2, BUTTON #4: LightHouse *****
on mouseUp
  go to first card of stack PMS_driver
end mouseUp

 ***** BKGND #2, BUTTON #5: up *****
on mouseDown
  put the scroll of field MIP_no_resp into it
  subtract 100 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of field MIP_no_resp to it
  set the scroll of field nomenclature_resp to it
  set the scroll of field ID_resp to it
  set the scroll of field location_resp to it
end mouseDown

 ***** BKGND #2, BUTTON #6: New Button *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #2, BUTTON #7: go_top *****
on mouseDown
  set the lockscreen to true
  set the scroll of field MIP_no_resp to 0
  set the scroll of field nomenclature_resp to 0
  set the scroll of field ID_resp to 0
  set the scroll of field location_resp to 0
  set the lockscreen to false
end mouseDown

 ***** BKGND #2, BUTTON #8: down *****
on mouseDown
  put the scroll of field MIP_no_resp into it
  add 100 to it
  set the lockscreen to true
  set the scroll of field MIP_no_resp to it
  set the scroll of field nomenclature_resp to it
  set the scroll of field ID_resp to it
  set the scroll of field location_resp to it
  set the lockscreen to false
end mouseDown

 ***** BKGND #3, BUTTON #1: New Button *****
on mouseUp
  ask "Search for what ?"
  if it is empty or it is "CANCEL" then
    exit mouseUp
  else
    put it into FIND_WHAT
    push card
    find FIND_WHAT
    if the result is not empty then
      beep
      answer "String not found"
      exit mouseUp
    end if
  end if
end mouseUp

 ***** BKGND #3, BUTTON #3: New Button *****
on mouseUp
  global TRANSFERED_DATA
  set cursor to 4
  put "PRINTED ON:" &"     " & the date &"     " & the time & return
  & return & return into TRANSFERED_DATA
  put "Personell available today for Work Center :" && fld WC_personel into temp
  put centered(temp,82)
  & return & return & return after TRANSFERED_DATA
  put "           Name                    Rank       Specialty Code    Position"
  & return & return after TRANSFERED_DATA
  repeat with i = 1 to the number of lines of fld ship_data_name
    put centered(line i of fld ship_data_name,26)
    & centered(line i of fld ship_data_rank,12)
    & centered(line i of fld ship_data_spec_code,20)
    & line i of fld ship_data_position & return after TRANSFERED_DATA
  end repeat
  push card
  lock screen
  go to stack generic_fld_print
  send mouseUp to bg button print_me
  pop card
  unlock screen
  put empty into TRANSFERED_DATA
end mouseUp

 ***** BKGND #3, BUTTON #4: LightHouse *****
on mouseUp
  go to first card of stack PMS_driver
end mouseUp

 ***** BKGND #3, BUTTON #5: up *****
on mouseDown
  put the scroll of field ship_data_name into it
  subtract 100 from it
  if it < 0 then put 0 into it
  set the lockscreen to true
  set the scroll of field ship_data_name to it
  set the scroll of field ship_data_rank to it
  set the scroll of field ship_data_spec_code to it
  set the scroll of field ship_data_position to it
end mouseDown

 ***** BKGND #3, BUTTON #6: New Button *****
on mouseUp
  go first
end mouseUp

 ***** BKGND #3, BUTTON #7: go_top *****
on mouseDown
  set the lockscreen to true
  set the scroll of field ship_data_name to 0
 `set the scroll of field ship_data_rank to 0
  set the scroll of field ship_data_spec_code to 0
  set the scroll of field ship_data_position to 0
  set the lockscreen to false
end mouseDown

 ***** BKGND #3, BUTTON #8: down *****
on mouseDown
  put the scroll of field ship_data_name into it
  add 100 to it
  set the lockscreen to true
  set the scroll of field ship_data_name to it
  set the scroll of field ship_data_rank to it
  set the scroll of field ship_data_spec_code to it
  set the scroll of field ship_data_position to it
  set the lockscreen to false
end mouseDown

***** CARD #1, BUTTON #1: active *****
on mouseUp
  hide me
  show card button inactive
end mouseUp

***** CARD #1, BUTTON #2: inactive *****
on mouseUp
  hide me
  show card button active
end mouseUp


SCRIPTS FOR STACK: spot_check
_______________________________________________________

****** STACK SCRIPT *****
on openstack
  if the freesize of this stack > 0.15 * the size of this stack then
    doMenu "Compact Stack"
  end if
end openstack
```

LIST OF REFERENCES

1. Giannotti, G. CDR. and Duffy, Kevin LT., *"ARGOS: Design and Development of Object-oriented, Event-driven, Multimedia Database Technology in Support of the "Paperless Ship"*, Masters Thesis, Naval Postgraduate School, Monterey, Ca., December 1988.
2. Goodman, D., *The Complete Hypercard^{tm} Handbook*, Bantam Computer Books, 1988.

BIBLIOGRAPHY

Apple Computer, Inc., *Apple: Technical Introduction to the Macintosh Family*, 1987.

Bond, G., *Xcmd's for Hypercard,* Management Information Source Inc., 1988.

Burke, H.E., *Handbook of Barcoding Systems*, Reinhold Company Inc., 1984.

Goodman, D., *Hypercard^{tm} Developer's Guide*, Bantam Computer Books, 1988.

Kaehler, C., *Hypercard Power - Techniques and Scripts*, Addison-Wesley Publishing Company Inc., 1988.

Knaster, S., *Macintosh Programming Secrets*, Addison-Wesley Publishing Company Inc., 1988.

Korth, H.F., Silberschatz, A., *Database System Concepts*, MacGraw-Hill Book Co., 1986

Maintenance Material Management Manual, OPNAVINST 4790.4 (Series), 1987.

Swaine, M., *Dr. Dobb's Essential Hypertalk Handbook*, M & T Publishing Inc., 1988.

Shafer, D., *HyperTalk Programming*, Hayden Books, 1988.

Waite Group, *HyperTalk™ Bible*, Hayden Books, 1989.

Waite Group, *Tricks of the HyperTalk™ Masters*, Hayden Books, 1989.

West J., *Programming with the Macintosh Programmer's Workshop*, Bantam Books, 1987.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Dudley Knox Library 2
Code 0142
Naval Postgraduate School
Monterey, California 93943-5002

3. Office of Reasearch Adiministration 1
Code 012
Naval Postgraduate School
Monterey, California 93943-5002

4. Chairman, Computer Science Dept. 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5002

5. Chief of Naval Research 1
800 N. Quincy Street
Arlington, Virginia 22217-5000

6. Center for Naval Analyses 1
4401 Ford Avenue
Arlington, Virginia 22302-0268

7. Naval Ocean Systems Center 1
271 Catalina Boulevard
San Diego, California 92152

8. Curriculum Officer 1
Computer Technology Program, code 37
Monterey, California 93943-5000

9. Professor C. Thomas Wu (Code 52Wq) 16
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

10. Maria M. Jamini-Ramirez 2
Division Head
MDS Division
Data Systems Department
Naval Weapons Station
Concord, California 94520-5000

11. Robert Calogero 2
Director SEA CEL-PA
Logistics Policy and Appraisal Division
Naval Sea Systems Command
Washington, D. C. 20362-5101

12. Clifford G. Geiger 1
Deputy Chief Engineer - Logistics
Naval Sea Systems Command
Washington, D. C. 20362-5101

13. Lieutenant H.V. Turner 2
8909 46th St. West
Tacoma, Washington 98466

14. Lieutenant D. G. Antonopoulos 2
1002 Pacific Grove Lane #1
Pacific Grove, California 93950

6/4-583

Thesis
A5657
c.1

Antonopoulos
Design and implementation of the PMS module for "ARGOS".

Thesis
A5657
c.1

Antonopoulos
Design and implementation of the PMS module for "ARGOS".

AMERICAN BINDERY
2
TOPEKA KANSAS

Design and implementation of the PMS mod

3 2768 000 87734 4
DUDLEY KNOX LIBRARY